1974

An aid for the allocation of resources in ship repairs at naval shipyards.

Medina Aedo, Enrique.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/16876

AN AID FOR THE ALLOCATION OF RESOURCES IN SHIP REPAIRS AT NAVAL SHIPYARDS

Enrique Medina Aedo

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN AID FOR THE ALLOCATION OF RESOURCES IN SHIP REPAIRS AT NAVAL SHIPYARDS

by

Enrique Medina Aedo

Thesis Advisor: F. R. Richards

September 1974

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Aid for the Allocation of Resources in Ship Repairs at Naval Shipyards		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis (September 1974)
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Enrique Aedo Medina		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1974
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Resource Allocation Naval Shipyards

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis provides shipyard planners with an aid to help them make their daily decisions about scheduling jobs and allocating manpower resources whilw trying to accomplish each project without delay beyond its scheduled completion date. A heuristic algorithm which focuses on the specific

problems of the shipyard planners is developed and a computer program is included which performs all the necessary calculations and gives the planners a daily assignment of resources.

Four other allocation procedures are surveyed. Two of these give solutions to the single project, multiresource problem; one procedure is an analytical model, the other is an empirical method; a third procedure is a heuristic approach to a multiproject, multiresource problem; and the last procedure is an analytical model which applies to the single project, single resource problem.

An Aid for the Allocation of Resources in Ship Repairs at Naval Shipyards

by

Enrique Medina Aedo
Commander, Navy (Chile)

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1974

Thesis
M4265
c 1

ABSTRACT

This thesis provides shipyard planners with an aid to help them make their daily decisions about scheduling jobs and allocating manpower resources while trying to accomplish each project without delay beyond its scheduled completion date. A heuristic algorithm which focuses on the specific problems of the shipyard planners is developed and a computer program is included which performs all the necessary calculations and gives the planners a daily assignment of resources.

Four other allocation procedures are surveyed. Two of these give sclutions to the single project, multiresource problem; one procedure is an analytical model, the other is an empirical method; a third procedure is a heuristic approach to a multiproject ,multiresource problem; and the last procedure is an analytical model which applies to the single project, single resource problem.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I wish to thank my thesis advisor, Professor Russell Richards for all the assistance he gave me during the preparation of this thesis. I also want to acknowledge the many beneficial comments of Professor Alan McMasters. Additionally, I thank Mrs. William Zeleny for the help she gave me during the development of the computer program .

I. INTRODUCTION

A. THE NAVAL SHIPYARD

The naval shipyard is a very complex activity, whose main task is to maintain the combat capability of a ship at its highest level by performing corrective and preventive maintenance. The jobs range from simple repairs to extensive overhauls of all the different systems.

The shipyard is composed of different divisions, each one specializing in the maintenance of a certain system of the ship. The components of these divisions are workshops which have the men and equipment necessary to fulfill the demands imposed by the jobs to be performed.

Since the complexity of a modern ship is so great,each workshop can be expected to undertake jobs in only a specialized area of a particular system. There is negligible overlap in the area of specialization between shops of the same division.Even within a workshop itself, the technicians will often be specialized in only one of the many possible technical aspects that may be the responsibility of the shop. For example,in the case of the Electronics shop some technicians must be experts in Radar Repeaters, others in ECM equipment,and so on.

Furthermore, teams of workers are often trained not only in certain specialized areas, but also for equipments for only a certain type of ship. Rarely does one find a workshop with two teams of workers that share exactly the same technical knowledge. This is especially true in shops with personnel having high technical levels in electricity, electronics or weapons systems. In other shops with less stringent preparation demands, such as piping or boilers, the degree of specialization is not so high ; generally, in

the latter case, the knowledge is more homogeneous and work teams within these shops may assist each other.

B. THE PROBLEM OF THE PLANNERS

When a ship arrives at the shipyard for repairs, every job that is to be done to her is analyzed to determine its estimated duration in man-days, the shop involved, and the material resources needed.This information is usually put in arrow diagram form and constitutes the project of all the work that must be done to the ship. This project is to be used by the planners who must assign the resources required day by day.

There will be occasions where two or more workshops will have to work in parallel, but most of the time one shop alone will have to work on an activity.

Since under normal circumstances more than one ship, in fact many ships, will be present at the same time for repairs, many projects have to be dealt with . Although it may appear that they are independent, there are many interactions between them because the utilization of one type of resource in one of them reduces the availability of that rescurce for the rest of the projects.

The projects have a definite due date which must not be surpassed once it has been set. Under normal circumstances the deadline must be met using resources only inside the shipyard without working overtime. It is therefore imperative that the available resources be used as intelligently and efficiently as possible.

The planners' task is a formidable one,and it is easy to see that PERT or CPM techniques alone, although very helpful,are not enough to enable the planners to cope with

the complexity of the assignment problem. If there are more jobs requiring a certain specialty than there are men available, the planners must determine which jobs, if delayed, will produce the least harm to the overall work.

Also, at the workshop level, the chief of the shop and his planners are faced with similar problems; they receive demands from higher echelons that certain jobs with fixed starting and finishing dates be carried out on specific ships. They have to assign the right men to each of the jobs for which their workshop is responsible while taking care that the crews assigned are sufficient so that the jobs are not delayed. This is especially crucial in those technical shops with highly specialized workers.

Since the number of planners is small and the amount of work is large, time is not available to do a detailed analysis of the optimal way to assign the workers to the various jobs. Consequently, these decisions are made in most cases, without any basis other than by experience. What is needed is some relatively simple-to-use technique to help the planners make their decisions in a short amount of time.

C. THESIS PREVIEW

In Chapter II, an empirical method for solving the problem of allocating resources in a single project, multiresource case is presented. In Chapter III, two analytical methods for solving the time/cost trade off problem are surveyed and in Chapter IV a computer program based on a heuristic approach for resource assignment on a multiproject problem is presented.

II. EMPIRICAL METHOD

A. INTRODUCTION

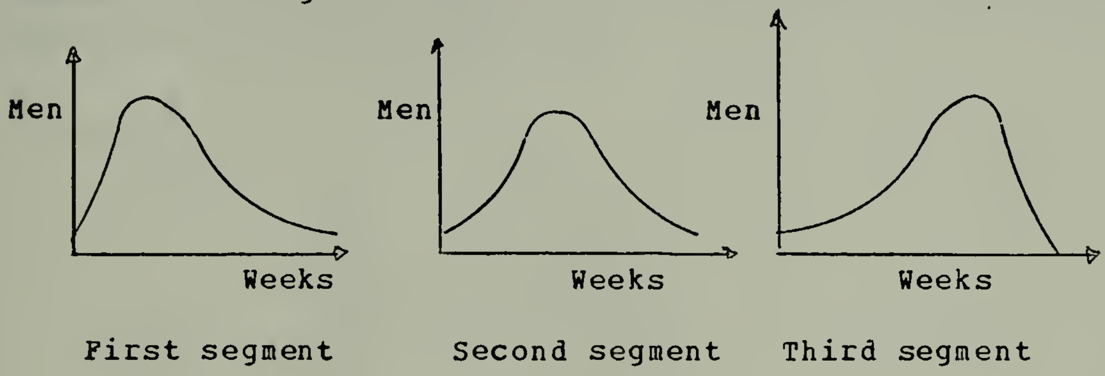
For a certain period of time in the past, before the sixties decade, very little work was done in the area of developing techniques for solving the problem of long range resource planning with limited resources. At that time the planners of some shipyards were using an empirical method based on curves obtained after years of observations and validation, to allocate resources to all the activities comprising a general repair project [Ref.1]. This method assumed ample resources available.

Although the solution obtained was not optimal, it provided the planners with some basis to allocate their men by specifying the amount of man-days to assign to each project on a weekly basis. Later, with the applicaticn of PERT to construction projects, the Empirical Method was refined to include this technique. As a consequence of this refinement, besides knowing the amount of man-days to allocate to a project, the workshop chiefs also had information about the distribution of the manpower among the different activities.

This refined empirical method is, unfortunately, restricted to the case of a single project with ample resources (man-hours) to accomplish all of the required work. There are more sophisticated analytical models available in the open literature which provide optimal solutions to this problem: a single project with ample resources. In the next section, the empirical method is presented, and two of the analytical methods are discussed in Chapter III, together with a heuristic method which can handle the mcre general multiproject, multiresource problem.

B. THE METHOD

At the arrival of a ship at the shipyard a decision is made about the time that ship will be scheduled for dry dock. Three curves have been determined empirically from the historical data concerning the average number of man-days necessary for each work week the ship is in the shipyard. The curve used for a particular ship depends on whether that ship,is scheduled for dry dock during the first, second,or third segment of its total repair duration time.
The shapes of the curves vary with the dry dock interval as shown in the figure below



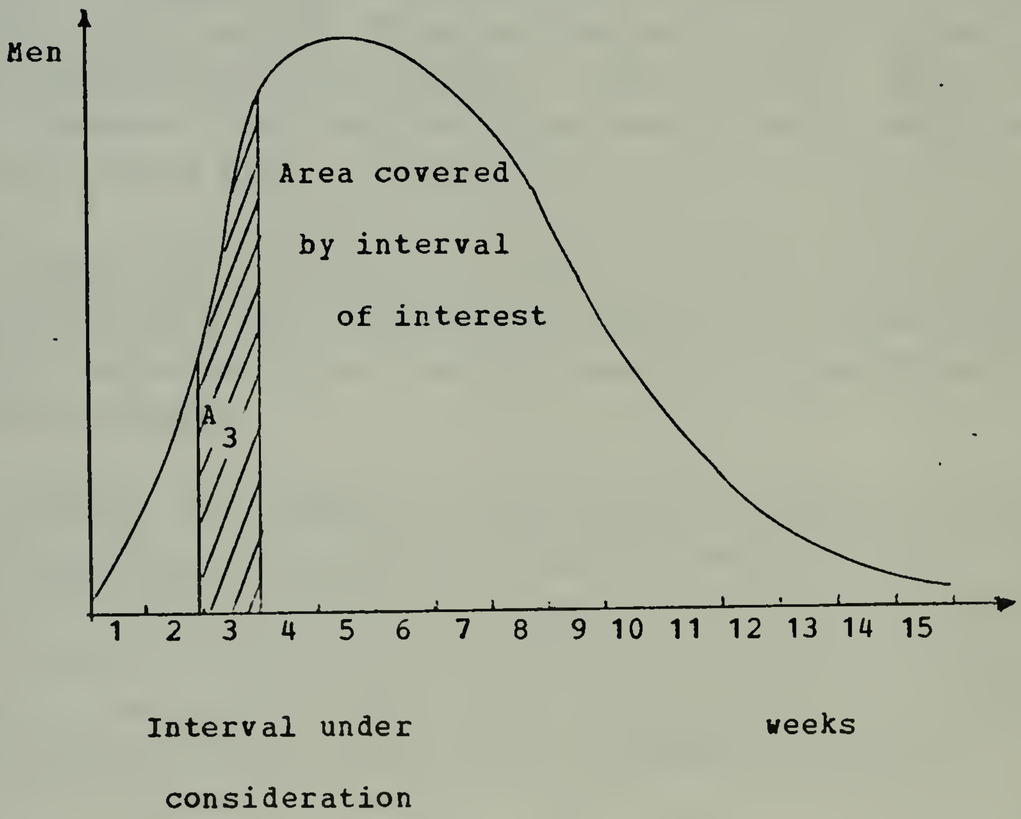
Curves for Resource Allocation

Fig.1

After the appropriate curve is selected, the base of the curve is divided into several intervals of equal length, where the number of intervals corresponds to the estimated number of weeks that will be required to complete the repairs. To obtain the amount of man-days to assign to a project, the planner goes to the week under consideration and determines the area of the curve over the appropriate interval. That area is then divided by the total area under the curve, and the resulting factor is used as the percentage of total manpower to be allocated to that particular ship.

This information is given to each shop where a predetermined factor is applied to the number received to obtain the number of man-days the shop must allocate during that week to the project. The factor is based on the percentage of the shipyard's total work force that is contributed by the shop.

The procedure is shown in the figure that follows. It has been assumed that the ship will go into dry dock at the end of the fcurth week of a total repair duration of 15 weeks. Since this is during the first third of the repair period, the first curve is used. The week under consideration is the third week.



Curve for first segment
Fig.2

The total area is normalized so that A_3 as given by the curve is directly in terms of percentage of the total amount of work, in man-weeks, to be done to the ship during the third week.

If every shop has a PERT diagram for each ship under repair, the number of men assigned to a given ship can be allocated to specific activities depending on which jobs are most critical (on the critical path).

When the number of ships under repair is high, the problem of assigning men to each activity every day becomes untractable unless the shop has a staff sufficiently large to keep the network diagrams updated from day to day, so that everything is under control. Since a large staff per shop increases the indirect labor costs prohibitively,this procedure could not be followed to the letter in practice.

Perhaps more important is the fact that the empirical procedure does not explicitly consider the dependencies between projects which result from the competition for limited resources.

Another shortcoming of the empirical procedure is that the allocation of resources is dcne on a weekly basis. This may greatly impair the flexibility of allocation , since a fixed amcunt of resources are assigned for all of a week's period tc one project even though another project may become more important due to some activity becoming critical.

These last two factors make strongly desirable a method that considers the case of multiresource, multiproject allocaticn on a day-by-day basis with provisions for the users at the shop level so that they have all the information they need without any further calculations.

III. RESOURCE ALLOCATION MODELS

A. INTRODUCTION

There are several different methods of solution to the problem of allocating resources in project scheduling. These differ by the degree of limitation that has been imposed on the resources.

When the resources are sufficiently large so that there are no conflicts in their usage by competing activities, all the activities may be scheduled at their early start time. For the case where resources are ample, the project completion time together with cost are the relevant aspects of the problem and time/cost optimization or time/cost tradeoffs are the objectives of the models.

Several analytical methods have been suggested for solving this case. Some of the solution techniques include PERT/COST,dynamic programming and linear programming [Ref.2]. All of the methods have been adapted for computer use.

If the resources are ample and it is possible to schedule each activity as before at its early start time, the objective generally turns to that of smoothing out the use of the resources while achieving the schedule completion date of the project. Through a 'leveling' of resource usage, labor costs are usually minimized because an attempt is made to avoid excesive hiring and firing costs and overtime costs [Ref. 2 and 5].

For the leveling problem, all of the solutions are of the heuristic type, ranging from the case of a single project with several types of resources (see, Burgess and Killebrew[Ref.3] and Wiest [Ref.4]) to a multiproject,

multiresource case (Levy,Thompson and Wiest[Ref. 5]).

A third type of problem arises when the resources are very limited, and there are conflicts in their use by the different activities involved. For this case there are even fewer solution methods offered than for the other two cases. Unfortunately, this is the real problem that normally faces a Naval Shipyard.

Shackelton [Ref. 7] presents two analytical methods for this type of problem. One is based on a mixed integer-linear programming model and the other based on a dynamic programming model. These two models are outlined in the following sections. The first model applies to the case of a single project with multiple resources, while the second only applies to the single project, single resource problem.

Following Shackelton's two models a heuristic approach to the more general problem of several projects and several resources is presented.This latter heuristic approach developed by Wiest [Ref.6] is most directly applicable to the problem facing shipyard planners. Therefore, it has been modified in this thesis to provide a heuristic procedure for solving the shipyard planning problem. A computer program was written to perform all the necessary calculations, and it is described in Chapter IV. The program is included in the section 'Computer Program'.

B. MIXED INTEGER LINEAR PROGRAMMING MODEL

1. General

This model was developed for the single project,multiresource case, where the resource profiles are fixed. For the solution algorithm, Shackelton exploits the similarity between this problem and the transportation problem. After converting the single project, multiresource scheduling problem with the objective to minimize the cost of allocating resources to a problem like the transportation problem, a partitioning procedure developed by Benders [Ref.8] is used to obtain the final solution.
A sketch of the problem formulation follows.

2. The Model

The problem is to minimize the total cost of assigning resources to the activities of the project subject to constraints which require that:
1) The man-days assigned to activities that need men from shop k, on day t, should not be greater than the men available at shop k for day t.
2) The men assigned to an activity for a certain period must equal the man-days needed to complete the job.
Mathematically,

$$\text{Min} \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{(i,j)\in S_k} C_t \, r_{ijt}^{k}$$

Subject to:

$$\sum_{(i,j)\in S_k} r_{ijt}^{k} \le R_t^{k} \qquad k=1,2,..,K \qquad t=1,2..,T$$

$$\sum_{t=1}^{T} r^{k}_{ijt} = M_{ij} \qquad k=1,2,\ldots,K$$

$$r^{k}_{ijt} \geq 0 \qquad \text{For all } i,j,k,t.$$

Where:

C_t =Cost on tth day

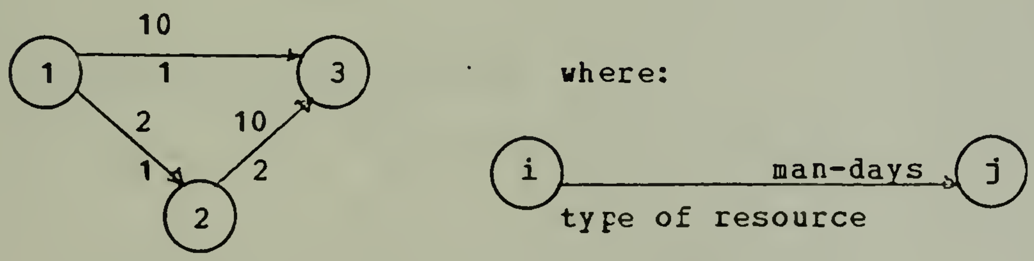
r^{k}_{ijt} =number of men from shop k, used on activity (i,j) on day t

S_k =Set of all activities (i,j) requiring men from shop k.

M_{ij} =man-days needed to complete activity (i,j).

R^{k}_{t} =men available at shop k on day t

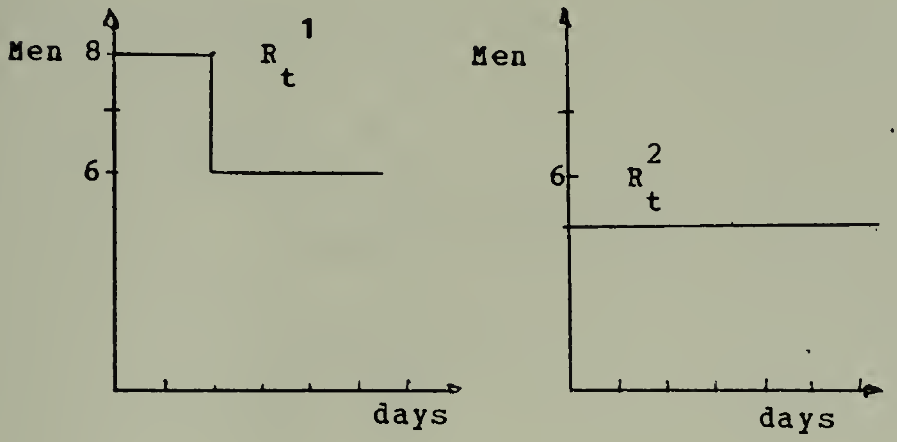
To demonstrate the equivalence between this problem and the transportation problem, a simple project with two resources will be used as an example.



Network for illustrative example

Fig.3

Let the time to finish the project be five days and let the resources for shop #1 and shop #2 be as shown on the following figures, for the five days considered.



R_t^i=men available at shop i on day t

Shop 1 Shop 2

Resource profiles

Fig.4

The objective function is:

$$\text{Min } C_1(r^1_{121}+r^1_{131}+r^2_{231})+C_2(r^1_{122}+r^1_{132}+r^2_{232})$$
$$+C_3(r^1_{123}+r^1_{133}+r^2_{233})+C_4(r^1_{124}+r^1_{134}+r^2_{234})$$
$$+C_5(r^1_{125}+r^1_{135}+r^2_{235})$$

S.T.

$$r^1_{121}+r^1_{131}\leq R^1_1=8$$

$r^1_{122}+r^1_{132} \leq R^1_2 = 8$ for shop 1

$r^1_{123}+r^1_{133} \leq R^1_3$

$r^1_{124}+r^1_{134} \leq R^1_4$

$r^1_{125}+r^1_{135} \leq R^1_5$

$r^2_{231} \leq R^2_1 = 5$ for shop 2

$r^2_{232} \leq R^2_2 = 5$

$r^2_{233} \leq R^2_3$

$r^2_{234} \leq R^2_4$

$r^2_{235} \leq R^2_5$

$r^1_{121} + r^1_{122} + r^1_{123} + r^1_{124} = M_{12} = 2$

$r^1_{131} + r^1_{132} + r^1_{133} + r^1_{134} = M_{13} = 10$

$r^2_{231} + r^2_{232} + r^2_{233} + r^2_{234} + r^2_{235} = M_{23} = 10$

and $r^k_{ijt} \geq 0$

To minimize project duration Shackelton suggests selecting values for C_t such that $0 \leq C_1 \leq C_2 \leq C_3 \leq C_4 \leq C_5$

This problem can be converted to the classical transportation problem of linear programming if we add to

the problem a nonnegative slack s_t^k for day t in shop k to each of the inequality constraints and the following equations associated with two dummy destinations.

$$s_1^1 + s_2^1 + s_3^1 + s_4^1 + s_5^1 = M^1$$

$$s_1^2 + s_2^2 + s_3^2 + s_4^2 + s_5^2 = M^2$$

where

$$M^1 = R_1^1 + R_2^1 + \dots R_5^1 - M_{12} - M_{13}$$

$$M^2 = R_1^2 + R_2^2 + \dots R_5^2 - M_{23}$$

The resulting problem can be written in matrix form:

$$\text{Min } CX_1 + CX_2$$

S.T.

$$\begin{array}{lll} A_1X_1 & = a_1 & \text{(Sources)} \\ B_1X_1 & = b_1 & \text{(Destinations)} \\ A_2X_2 & = a_2 & \\ B_2X_2 & = b_2 & \\ X_1 \geq 0 & & \\ X_2 \geq 0 & & \end{array}$$

where

$$CX_1 = \sum_{T=1}^{5} C_t (r_{12t}^1 + r_{13t}^1)$$

$$CX_2 = \sum_{t=1}^{5} c_t \, (r^2_{23t})$$

$$A_1X_1 = \begin{bmatrix} r^1_{121}+r^1_{131}+s^1_1 \\ r^1_{122}+r^1_{132}+s^1_2 \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ r^1_{125}+r^1_{135}+s^1_5 \end{bmatrix} \qquad a_1 = \begin{bmatrix} R^1_1 \\ R^1_2 \\ R^1_3 \\ R^1_4 \\ R^1_5 \end{bmatrix}$$

$$B_1X_1 = \begin{bmatrix} r^1_{121}+r^1_{122}+\ldots+r^1_{125} \\ r^1_{131} \; r^1_{132}+\ldots+r^1_{135} \\ s^1_1 \; +s^1_2 \; +\ldots+s^1_5 \end{bmatrix} \qquad b_1 = \begin{bmatrix} M_{12} \\ M_{13} \\ M^1 \end{bmatrix}$$

$$A_2X_2=\begin{bmatrix} r^2_{231}+s^2_1 \\ r^2_{232}+s^2_2 \\ \cdot \quad \cdot \\ \cdot \quad \cdot \\ r^2_{235}+s^2_5 \end{bmatrix} \qquad a_2=\begin{bmatrix} R^2_1 \\ R^2_2 \\ R^2_3 \\ R^2_4 \\ R^2_5 \end{bmatrix}$$

$$B_2X_2=\begin{bmatrix} r^2_{231}+r^2_{232}+\dots+r^2_{235} \\ \\ s^2_1 \quad +s^2_2 \quad +\dots+s^2_5 \end{bmatrix} \qquad b_2=\begin{bmatrix} M_{23} \\ \\ M^2 \end{bmatrix}$$

In addition to the constraints above, the network problem also needs a requirement that the logical sequence of the activities be preserved. Also, a constraint is required to guarantee that when an activity is started it may not be interrupted.

Suppose activity (1-2) takes two days to complete. Then, using the fact that activity (1,2) must precede activity (2,3), the constraints are:

$r^1_{121}>0$

$r^1_{122}>0$

$r^1_{123}=0$

$r^1_{124}=0$

$r^1_{125}=0$

and $r^{2}_{231}=0$

$r^{2}_{232}=0$

For the second requirement that activities underway not be interrupted, the following constraints must be added:

if $r^{1}_{122}>0$

and $r^{1}_{123}=0$

then $r^{1}_{124}=0$

$r^{1}_{125}=0$

Shackelton takes care of these constraints by adding two,zero-cne, variables G and D, whose values depend on the status of the activity

Activity	G	D
Not started	0	0
In process	0	1
Finished	1	0

In the example, using the condition that activity(1-2) must precede activity(2-3) the constraints are:

$$r^{1}_{121}+r^{1}_{122}+r^{1}_{123}+r^{1}_{124}-(M_{12}-1)\,D_{124}-M_{12}\,G_{124}<0$$

$$r^{1}_{121}+r^{1}_{122}+r^{1}_{123}+r^{1}_{124}-D_{124}-M_{12}\,G_{124}>0$$

$$D_{124}+G_{124}<1$$

$$r^{2}_{235}-M_{23}G_{124}<0$$

The abcve inequalities state that the total allocation in the first four days is equal to zero(activity not started), greater than zero but less than M_{12}(activity in process) or equal to M_{12}(activity completed); and that activity (2-3) may not have any allocation until activity (1-2) is completed

To provide for the case when activity(1-2) is not finished in period 4, an additional allocation is required in period 5

$$-D_{123}+D_{124}>0$$

The entire set of constraints can be represented in matrix form:

$$D_1X_1+D_2X_2+Gy>d$$

where D_i :matrix of coefficients of the allocations

G :matrix of coefficients of 0-1 variables

y :Column vector of 0-1 variables

d :R.H.S. of constraints

The complete problem becomes:

$$\min \quad C_1X_1+C_2X_2$$

S.T.:

$$
\begin{aligned}
A_1X_1 &= a_1\\
B_1X_1 &= b_1\\
A_2X_2 &= a_2\\
B_2X_2 &= b_2\\
D_1X_1+D_2X_2+Gy &> d\\
x_i &> 0
\end{aligned}
$$

The problem is solved using Benders' partitioning procedure [Ref.8]. One should observe that even very simple single project, multiresource problems require a large number of constraints using this algorithm. This is an obvious disadvantage of the algorithm which would make the solution of a very complex problem extremely tedious.

C. DYNAMIC PROGRAMMING MODEL

1. General

The dynamic programming model was developed for the case of a single project with a single resource. The assumptions are that a single resource is required by all the activities in the project, overtime is prohibited and events can be ordered so the precedence relations are maintained.

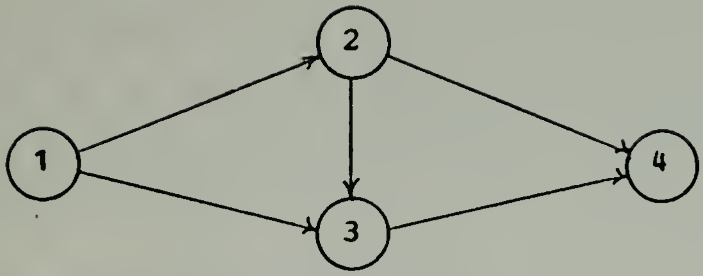
Once the problem is formulated as a dynamic programming problem with the stage returns and stage transformations established, Bellman's Principle of Optimality [Ref. 9], is used to obtain an expression for the project duration. The problem then reduces to a linear programming problem with the objective to minimize project duration.

2. The Model

The problem is stated as follows: given the resources available for each time interval and the requirements of all the activities of the network in terms of normal time for completion, find the assignment of resources which minimizes the cost.

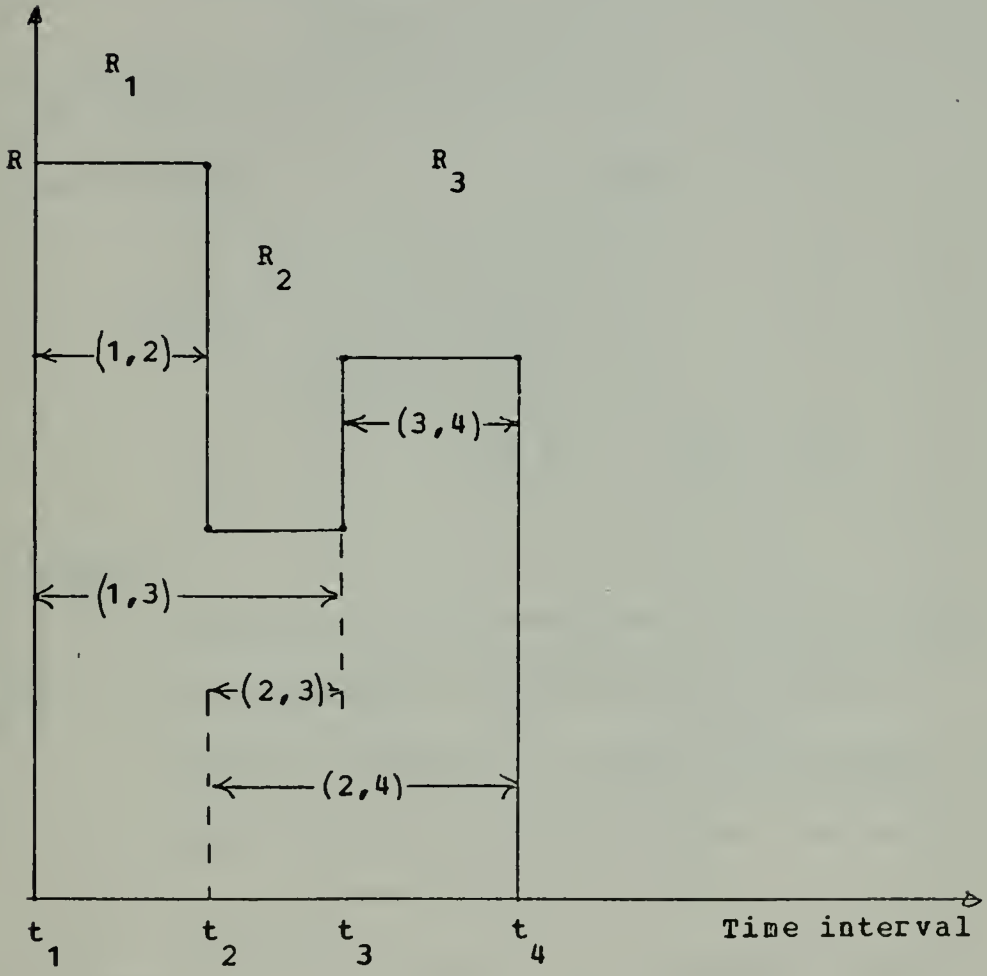
A simple project is again used as an example to show the development of this model.

The network and the resource profile are shown below:



Network used in example

Fig.5



Resource profile for the 3 periods considered.
(i,j) represent activities.

Fig.6

The problem is:

$$\text{Min } C_{12}r_{12}(t_1)(t_2-t_1) + C_{13}r_{13}(t_1)(t_2-t_1) + C_{13}r_{13}(t_2)(t_3-t_2)$$
$$+ C_{23}r_{23}(t_2)(t_3-t_2) + C_{24}r_{24}(t_4-t_3) + C_{24}r_{24}(t_4)(t_5-t_4)$$
$$C_{34}r_{34}(t_4)(t_5-t_4)$$

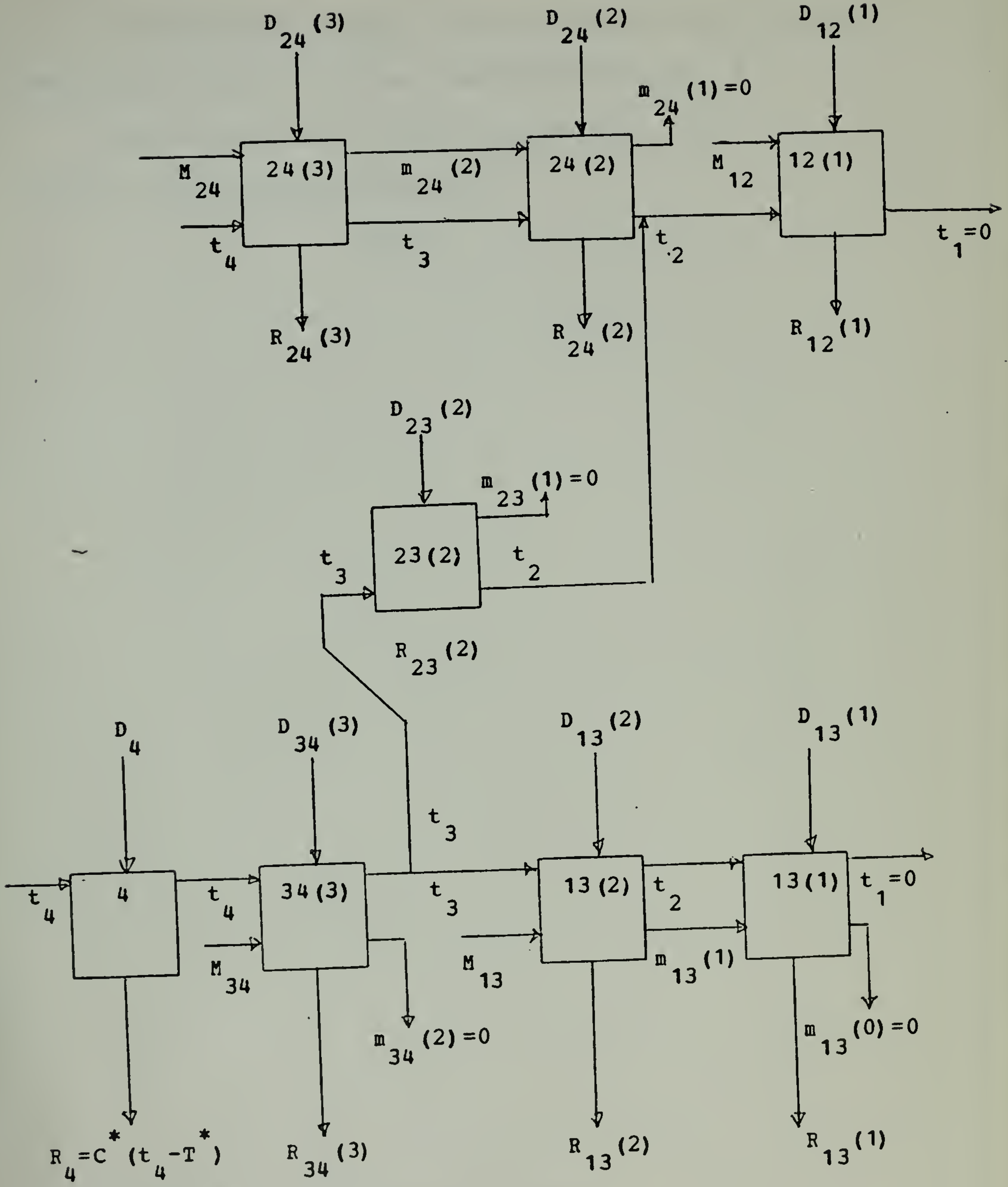
S.T.

$$r_{12}(t_1)(t_2-t_1\quad) = M_{12}$$
$$r_{13}(t_1)(t_2-t_1)+r_{.13}(t_3-t_2) = M_{13}$$
$$r_{23}(t_2)(t_3-t_2) = M_{23}$$
$$r_{24}(t_3)(t_4-t_3)+r_{24}(t_4)(t_5-t_4) = M_{24}$$
$$r_{34}(t_4)(t_5-t_4) = M_{34}$$
$$r_{12}(t_1)+r_{13}(t_1) \le R_1$$
$$r_{13}(t_2)+r_{24}(t_2)+r_{23}(t_2) \le R_2$$
$$r_{24}(t_3)+r_{34}(t_3) \le R_3$$
$$r_{ij}(t_a) \ge 0 \qquad a=1\ldots j-1$$

Where:

C_{ij}	=cost per man-day assigned to activity(i,j)
$r_{ij}(t)$	=men assigned to activity(i,j) on day t
M_{ij}	=man-days required to complete activity(i,j)
R_i	=number of men available during time period (t_i, t_{i+1})

This problem, can be represented as the block diagram shown below.



Block diagram for allocation problem

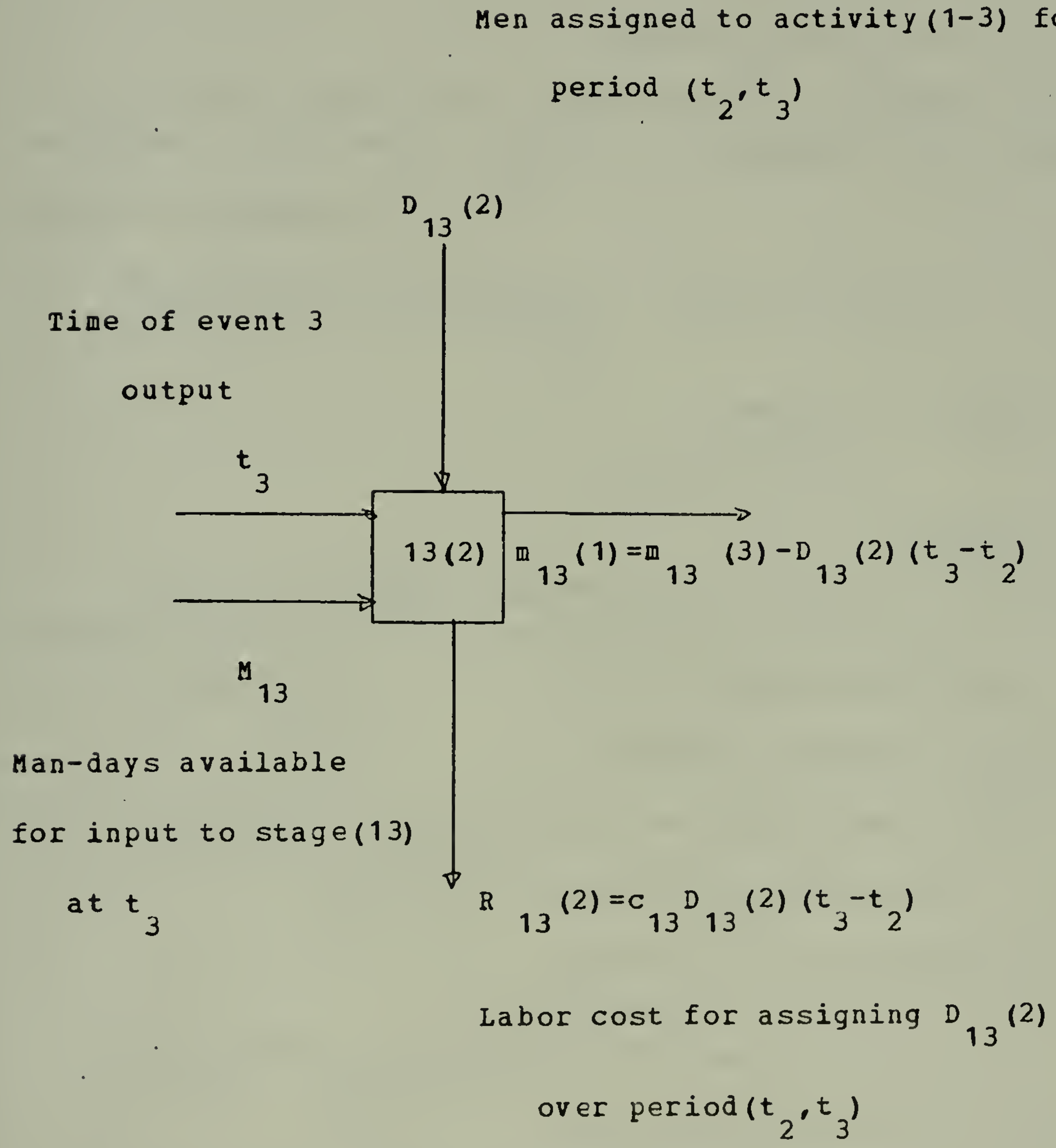
Fig.7

The decision $D_{ij}(k)$ is the number of men assigned to the activity(i,j) over the time interval(t_k, t_{k+1})

The stage return $R_{ij}(k) = C_{ij} D_{ij}(k)(t_{k+1} - t_k)$ and the stage transformation is:

$m_{ij}(t_k$

Figure 8 presents one of the blocks from the diagram of figure 7 and will be used to explain the expressions in figure 7.



Men assigned to activity (1-3) for period (t_2, t_3)
$D_{13}(2)$
Time of event 3 output
t_3
13(2)
$m_{13}(1) = m_{13}(3) - D_{13}(2)(t_3 - t_2)$
M_{13}
Man-days available for input to stage (13) at t_3
$R_{13}(2) = c_{13} D_{13}(2)(t_3 - t_2)$
Labor cost for assigning $D_{13}(2)$ over period (t_2, t_3)

Fig. 8

The recursive equation for stage 13(1) is

$$f_{13}(1)(m_{13}(1),t_1,t_2)=\underset{D_{13}(1)}{\text{Min}}\ R_{13}(1)(m_{13}(1),t_1,t_2,D_{13}(1))$$

S.T.

$$D_{13}(2)=\frac{M_{13}(2)-m_{13}(1)}{t_3-t_2}$$

or

$$m_{13}(0)=m_{13}(1)-D_{13}(1)(t_2-t_1)$$

At this stage all the man-days available must be assigned and hence $m_{13}(0)=0$. Therefore the recursive equation becomes:

$$f_{13}(1)(m_{13}(1),t_1,t_2)=\underset{D_{23}(1)}{\text{Min}}\ C_3 D_{13}(1)(t_2-t_1)$$

S.T.

$$D_{13}(1)=\begin{cases}\dfrac{m_{13}(1)}{t_2-t_1} & \text{for } t_2>t_1\\ 0 & \text{otherwise}\end{cases}$$

Solving the dynamic programming problem, the general expression for the stage decision is found to be:

$$D_{ij}(i)=\frac{m_{ij}(i)}{t_{i+1}-t_i}$$ for the block where the activity begins at event i

$$D_{ij}(k)=\frac{m_{ij}(k)-m_{ij}(k-1)}{t_{k+1}-t_k}$$ for the block where the activity(i,j) is under process at an intermediate stage k

$$D_{ij}(j-1)=\frac{M_{ij}-m_{ij}(j-2)}{t_j-t_{j-1}}$$ for the block where the activity terminates at event j

After some manipulations to obtain the stage transformation for the event times, Shackelton determines an expression fcr project duration:

$$t_n > \sum_{(i,j)} \sum_{a=1}^{j-1} \frac{x_{ij}(a)}{R_a} :$$

where:

$$x_{ij}(k) \quad = m_{ij}(k) - m_{ij}(k-1) \qquad k=i,i+1,.j-2$$

$$x_{ij}(j-1) = M_{ij} - m_{ij}(k-2)$$

For the example given,the project duration equation is:

$$t_4 > \frac{X_{12}(1) + X_{13}(1)}{R_1} \; \frac{+X_{13}(2) + X_{23}(2) + X_{24}(2)}{R_2} + \frac{X_{34}(3) + X_{24}(3)}{R_3}$$

To obtain the minimum project duration subject to resource restrictions, the following linear programming problem must be solved:

$$\text{Min} \frac{(X_{12}(1) + X_{13}(1))}{R_1} + \frac{(X_{13}(2) + X_{23}(2) + X_{24}(2))}{R_2} + \frac{(X_{34}(3) + X_{24}(3))}{R_3}$$

S.T.

$$X_{12}(1) = M_{12}$$

$$X_{13}(1) + X_{13}(2) = M_{13}$$

$$X_{23}(2) = M_{23}$$

$$X_{24}(2) + X_{24}(3) = M_{24}$$

$$X_{34}(3) = M_{34}$$

$$X_{ij} \geq 0$$

The final solution procedure is a relatively simple linear programming application. The shortcoming is that the algorithm only applies to a single project with a single resource, and as such, is not generally useful to shipyard

planners. The third model that is investigated is applicable to the multiproject,multiresource problem. It obtains this generality at the expense of not being able to claim optimality. Being a heuristic algorithm, no promises of a best solution can be given, but good solutions can be expected.

D. HEURISTIC MODEL

1. General

This heuristic allocation model was developed by J.D.Wiest [Ref.6] for the case of several projects containing activities that compete for several types of limited resources.

In theory there is no limit on the number of projects it can handle,nor on the number of different resources that may be considered, but in practice limits are imposed by the capacity of the user to deal with many projects and resources at the same time and by the capacity of the computer.

2. The Allocation Procedure

The resources available are assigned, day-to-day, to those activities which are active, that is , those activities that may be scheduled for resource assignment, or which become active at the time considered. These activities are ordered by their early start times , based on a normal crew size ,with first priority for consideration of allocation of resources being given to the critical activities, which are those activities that have zero or negative slack. The noncritical activities constitute a second list which is considered when the critical activities have had their manpower demands satisfied.

Within each list, the activities are selected at random , the probability that the first activity of the list will be selected is p, where p is an input parameter between 0 and 1. If the first activity is not selected, the next activity of the list is considered with the same probability of selection, and so on until one of the activities is chosen. Those activities which were not chosen in the scan

wait until all the activities of the list have been tested before being considered again for selection.

Using this random selection procedure, replications can be made to obtain alternative schedules for the program. By scanning the alternate schedules, a planner can select that one which is considered best. If the number of replications is large, there should be a good chance that the schedule selected as best is nearly optimal. Without searching through all permutations of activities, (a mathematically infeasible task with several large projects) one cannot be sure that this schedule is optimal.

Four different types of crew -sizes are used in the allocation of the resources:minimum,normal,maximum and critical crew-size. Minimum crew-size is the minimum number of men that can be assigned to a job ;normal crew-size is the normal or usual number of men assigned to a job;maximum crew-size is the maximum number of men that can be assigned to a job without producing interference problems;critical crew-size is that crew size which when assigned to a job reduces its duration by one day.

The procedure first attempts to assign the critical crew size to the critical jobs. If this cannot be done, the algorithm tries to assign the normal crew size; before trying a minimum crew size, during the first scheduling attempt on the critical jobs,when normal crew size is not available , a procedure called Reschedule Active Jobs is used. This procedure scans the list of the active jobs and selects those that could have been scheduled to start at a later time,without affecting the project completion date. If the minimum crew size cannot be allocated to the activity, the algorithm searches the active jobs already scheduled to see if men can be 'borrowed' from them in sufficient quantity to schedule the new activity without

producing a delay on the overall project completion date.

If these two procedures still do not provide enough resources to start the critical job, even with minimum crew size, then the job is postponed one day.

The active noncritical jobs receive a more drastic treatment in that only two crew sizes are considered for their scheduling: normal and minimum crew sizes. Also, no second attempts are made to get them underway when the minimum crew size cannot be obtained from the resources available.

When a job requires more than one type of resource it is treated as a case of strictly parallel arcs,where each arc represents an activity using one resource,with the constraint that all of the activities incident from the node where the original job originates must start at the same time.

There are two other procedures which modify the procedure just described. The first of these tries to assign to each active critical job a maximum crew size to speed up those activities. Before starting the scanning procedure to select the activities to be scheduled on a given day, all the critical activities that had been assigned a crew size less than the maximum the day before have their crew assignments increased to the highest limit if resources are available for the given day.

The Add-on Unused Resources procedure compiles a list of those resources which were left unused at the end of the scheduling procedure, orders them by type and produces a list of active jobs which may receive these resources. The jobs are listed by ascending order of their total slack,and each one of them receives extra manpower until the unused resources are exhausted or until the list of jobs is

finished.

To have an evaluation of each alternative schedule produced, schedule-related costs are introduced in the model and an attempt is made to minimize total resource costs together with completion time. The expression for total cost is taken to be:

$$\text{Total Cost} = cz + \sum_{s=1}^{m} q_s^* w_s z .$$

where

c=Average cost (overhead expenses and/or due date penalties) in dollars per day.

z=Length of the schedule.

q_s^*=Max. number of men available in shop s.

w_s=Average wage in shop s, in dollars per day.

m=Number of shops.

In some cases the cost of increasing the shop resources above the normal level is less than the cost resulting from due date penalty and/or overhead charge whereas,in other cases the contrary is true. This led Wiest to a search procedure to seek some optimum of shop resource levels and resulting finishing date.

One search procedure starts with a minimum of resource levels, just sufficient to satisfy the needs of the most demanding activity. Once a schedule has been obtained together with its associated cost, the resources which are used by the critical jobs are increased by a certain amount. A new schedule with its cost is obtained and the rescurces are again increased, and the procedure is repeated as long there is some improvement in schedule cost.

The other search procedure starts at the other extreme

with resource levels that are such that all activities may start at their early start times, based on normal crew sizes.

The resource levels of all the shops are decreased and later, when no more changes are noticed, the resource level of one shop at a time is decreased until no more improvements are obtained.

The heuristic algorithm developed by Wiest applies to the multiproject,multiresource problem-the general type of problem that faces shipyard planners. Nevertheless, for the shipyard scheduling problem, there are some shortcomings to Wiest's procedure. In the first place, the shipyard planning problem is a dynamic problem. Ships move in and out of the shipyard,often on short notice. Because of this, the planners must be able to revise their schedules on a day to day basis. What good, for example, is a plan of allocating resources for a fixed set of ships and a given time interval if a new ship which must compete for those same resources enters the shipyard?

Ancther problem surfaces whenever unforseen delays cause jobs to require more man-hours than expected. In reality, all of the maintenance times are random variables. If they differ substantially from what the planners forecast, schedules can easily get messed up. A delay anywhere in the network can have a domino effect on the other unfinished activities. Planners must be able to react quickly to prevent this.

Finally, Wiest's algorithm takes as its objective to minimize total cost. This requires reliable estimates cf the various costs involved such as labor costs,overhead expenses, overtime costs and penalties. These costs are often difficult to determine. Additionally, most shipyards operate with a relatively fixed labor force. Workers must be

paid and overhead expenses incurred regardless of the utilization of the workers. Thus, it might be more appropriate, for the shipyard problem, to attempt to minimize project duration subject to the available resources and to ignore the costs.

The method developed in Chapter IV modifies Wiest's heuristic algorithm in an attempt to reduce the impact of problems such as those mentioned above.

IV. A COMPUTER PROGRAM FOR RESOURCE ALLOCATION

A. GENERAL

Since Wiest's model is the one that most closely represents the events that take place in a shipyard,it was chosen as a basis for the following Computer Program devoted to the problem of scheduling the repairs of ships with limited resources and several projects active at the same time.

In this Computer Program the unit of time is one day and for the computation of activity duration t given a crew assignment and the amount of work required,times are rounded to the nearest integer number of days. This approximation is believed to be especially valid in shipyard work, where the estimated amount of work required for a job (in man-days) is considered to be good if the error is not larger than 10%. Futhermore, work policies are such that if a job is finished before the end of the day a crew will not be reassigned to a new job unless they happen to finish before lunch.

As another departure from the models presented above, it is assumed that the total level of resources in a shipyard cannot be modified drastically and suddenly.Firing of people is not possible and hiring crews for short term specialized jobs is usually not permitted. Thus, crashing activities to speed-up a project at a higher cost by hiring more resources is not considered.

Finally, unlike Wiest's algorithm, the four crew sizes-minimum, normal, critical and maximum- are used only as starting points in the allocation of workcrews to activities. Actually, any integer number of workers between the minimum crew size and the maximum crew size may be assigned to an activity. This additional flexibility is more

representative of the actual practice in the shipyards.

B. DESCRIPTION OF THE PROGRAM

1. General Concepts

The Computer Program,written in Fortran IV, permits the allocation of resources, on a day-by-day basis, in problems involving several projects running in parallel, several types of resources employed by the different activities that constitute each project, restricted availability of these resources and a changing resource profile. The objective is to minimize project duration.

In one form of utilization of the program the activities selected for crew assignment are chosen on the basis of their early start times, based on normal crew size, with the critical jobs having priority. The second form uses a random selection of the active jobs.Several schedules are obtained from different selections of random numbers,and the user may choose the one which gives the minimum project duration.

The random selection mode requires greatly increased computer time over that required with the first mode . Also, it loses the flexibility of the day to day updating which is usually desirable as the shipyard dynamics change.

In both cases, a printed output is obtained for each day which indicates the activities that should be scheduled and the resources that should be allocated. Also, listings of amounts of resources that were not used and the status of all the activities for all the projects are presented.

2. The Input

The main input to the program is the node-arc incidence matrix of all the independent projects that are involved in the planning problem; the activity duration for the node-arc incidence matrix is determined on the basis of normal crew size used in each activity. This requires that all the projects must be written in the form of a PERT network, (see Moder and phillips [Ref. 10]). each node numbered following Fulkerson's algorithm (Appendix A) and using the activity-on-arc diagram.To arrive at this type of diagram from the precedence relationships, it is suggested that the user proceed from the precedence relationships to the activity-on-node diagram, and then from this type of diagram to the activity-on-arc diagram using an algorithm developed by J.H.Cyr (Appendix B).

As a secondary input, all the parameters that completely affect an activity's duration should be entered into the program.Some of these parameters will not change during the entire program run, some may vary due to changes in external conditions and some vary day by day due to the fact that activities are being scheduled every day,become active, become critical,need smaller crews or are completed. As already mentioned, the dynamic nature of the shipyard problem may require other changes.This input is explained in more detail on the section devoted to the user, where all the required parameters are discussed.

3. The Program

At the beginning the program computes all the data needed for a PERT/TIME network using the subroutine EWFWD. This subroutine scans each node-arc input matrix and selects all the non-zero elements of the matrix. First, a forward pass is performed and the early start (ES) times and early

finish (EF) times are calculated for each activity in the network. When the sink node is reached, a backward pass determines the latest finish (LF) time and the latest start (LS) time for each activity. The correponding slack for each activity is also determined. The results are printed under the heading, STATUS OF PROJECTS.

In the second step, each project is considered separately and those activities that will become active on the day under consideration are determined, are updated and added to the list of activities that were active on the previous day. Using the date input by the user, the program examines all the active jobs that had resources assigned the previous day. With this information it updates the amount of work remaining to be done (HH) which is, in turn, used to update LF and EF and to obtain the new slack. After updating all active jobs, the non-active jobs on the previous day are scanned to determine if their ES times are such that the jobs can be started. If they can be started, a further check is made to see if all of the incident activities into the starting node of the job under consideration have been completed. If this is the case the activity is declared active and its parameters are updated. If either condition is not fulfilled the activity remains non-active.

In all succeeding steps the projects are integrated and are considered as one large project by the program.

The active jobs, which are the only ones considered for crew assignment, are divided into two categories according to their total slack: active critical jobs and active non critical jobs.

The active critical jobs are given preferential treatment in the assignment of the crews. In assigning resources these jobs are ordered according to ascending early start times,if there is a tie in the early start times

the jobs are placed in ascending order with respect to activity duration.

Once the ordering has been established the program uses the Subroutine ASSCR (from ASSign to CRitical jobs) to fulfill the manpower needs of the jobs.

At this stage the user must choose the method the program is to use to select each job. Two methods are available:

a) deterministic selection of jobs.

b) random selection of jobs.

The deterministic alternative begins the resource allocation with the first activity on the list and works consecutively through the list. This alternative allows for daily updates, thus enabling shipyard planners to keep pace with the dynamic nature of shipyard requirements.

The random selection alternative selects jobs using a Monte Carlo method via the subroutine STOC. This alternative produces several alternative schedules, the number of replications controlled by an input parameter. The planner then can view the output for each replication and select the schedule yielding the shortest total project duration. As with Wiest's algorithm, the random selection alternative does not allow the planners to adjust the schedule on a day to day basis as delays occur or as the projects change.

For each of the critical jobs being processed the program tries to assign a critical crew size, which cannot be larger than the maximum crew and which size is such that it decreases by one day the total duration of the activity.

If not enough resources are available, a normal crew size is tried, and then a minimum crew size if everything else fails. If the minimum crew size cannot be allocated to a critical job, the program resorts to the BORROW Subroutine in a last effort to prevent delaying the job for a single

day.

The BORROW Subroutine , explained in detail in Appendix D, searches among all the active critical jobs that have resources assigned to find those which can give away part of their rescurces without decreasing their total slack. If the attempt to borrow resources fails, the complete project must be delayed at least one day.

On the other hand, if an activity obtains the critical crew size needed its early finish time is improved by one day or mcre. The program then goes back again to the BWFWD Subroutine to recalculate the parameters of the PERT network, since in this case the termination date has been advanced by one day,and it is possible that a new critical path has been created.

The treatment given the active noncritical jobs is different frcm the treatment given to the critical jobs, because, for noncritical jobs, the termination date for the project involved may not be affected by delays.

The ncncritical jobs are placed in ascending order of early start times based on normal crew size, then acccrding to ascending order of total slack if a tie should occur in the first parameter considered,and lastly they are ordered by activity duration if the total slacks are equal. After this ordering the program tests if there are still resources available once the crew assignments for the critical jobs have been completed. If resources are available the program uses ASSNC Subroutine (from ASSign to NonCritical jobs) to process these activities.

Again the method of activity selection may be deterministic or random,depending on the previous choice made by the user.

For the activity selected an attempt is made to assign

a normal crew size. If the resources available do not permit such an assignment, a minimum crew size is tried. If this fails, the activity is left in stand-by status with no resources assigned until the next day.

The program repeats these steps until all the active noncritical activities of the list have been processed. At this point a printout is produced indicating the status of all the active jobs,critical and non critical.

To ensure that all the available resources are used, the program goes through a final subroutine, the EXHST Subroutine (from EXHauST Resources) , explained in more detail in Appendix E, where any resources not used are assigned to active noncritical jobs which have already a ncrmal or minimum crew size assigned.

This step ends the resource allocations for a given day. The process is repeated for the following days until all activities in all projects have been scheduled.

C. IMPLEMENTATION

The program should be used on a daily basis, because it is felt that this option is more appropriate than the random selection option for the actual dynamic situation that faces the planners. Some of the scheduled activities will not follow exactly what was forecasted because of unforseen circumstances. Or, perhaps, some input parameter may vary from day to day requiring a daily correction.

In this section a detailed description of the input variables is given. These variables are classified as Fixed Inputs and Variable Inputs.

1. Fixed Inputs

The fixed inputs are those inputs which do not change during the entire program. The following variables are the fixed inputs:

NRES :Number of resources.

NP =Number of projects.

NRUNS:Number of replications(for the random selection alternative).

KST :A parameter for selecting the deterministic/random alternative.

LST :The total number of activities.

Also ,among the fixed inputs are those which specify the parameters necessary for the description of each project. In this group are:

LUT(K) :Number of arcs in project K

CMIN(I,J):Minimum crew size for activity(I,J).

CNOR(I,J):Normal crew size for activity(I,J).

CMAX(I,J):Maximum crew size for activity(I,J).

CODE(I,J):Type of resource needed by activity(I,J).

D(I,J) :Duration of activity(I,J).

The variables KST and NRUNS are used together for specifying the alternative, either deterministic or random. If the dterministic alternative is selected KST must be set to zero and NRUNS must also be set to zero; when the random alternative is selected KST must be set to one and NRUNS should be set to the number of replications of the complete scheduling process that is desired.

D(I,J), the duration of each activity is the required number of days assuming a normal crew size. The data are input as a square matrix corresponding to the node-arc diagram for each project. The rows represent the starting nodes for the activities and the columns represent the finishing nodes. For those pairs (i,j) which do not correspond to actual jobs,D(I,J) is set to zero.

It is also important that the value given to the variable CODE(I,J), which indicates the type of resource needed by activity(I,J),be an integer between one and NRES and that care is taken to be consistent in the use of each value of CODE to refer to a particular resource.

2.Variable Inputs

Variable Inputs are those inputs that change during the execution of the program.

The first variable input is the Julian date expressed by the variable T. The user must update this value each day.

Alsc included among the variable inputs are those parameters that change day to day because of the scheduling of activities that takes place in the program. The necessary parameters are:
CR(I,J):The crew size assigned to job(I,J).
HH(I,J):The duration of job(I,J) in man-days.

SW(I,J):The status of activity(I,J)--active or non active.

The value for HH(I,J) may be the same as the value obtained from the printout for the day, if there is agreement between what is forecast and what is actually going on with respect to that particular activity. On the other hand, if the user realizes that the estimation of HH(I,J) is not correct or that the real work accomplished during the day is different from that forecast, then the variable could be changed accordingly.

The same comments apply to the variable CR(I,J). The variable SW(I,J) normally should be obtained from the last day's printout.

Finally, there is the variable AV(I,J), which is the quantity of resource which is available. This variable may vary considerably so the user may want to update this information every day to keep the program up to date with the real existing conditions.

V. CONCLUSIONS

In a shipyard the shop planners need planning aids to help them make decisions about how to allocate their manpower resources each day while trying to maintain all job completion dates without delay. Or, provided sufficient resources are not available to perform all the required work, they need to determine what jobs should be postponed so as to produce the smallest overall delays. The computer program is written to take care of this particular problem, and has been designed with the idea that the user will be the shop planners. The objective is to provide the users with complete information each day about where to allocate each shop's manpower and in what quantity.

The program has no limitation on the number of projects, nor on the number of resources that it can handle (provided the problem remains within the capacity of the computer). The program is based on a heuristic algorithm, and as such can make no claim of optimality. Nonetheless, the approach is intuitively appealing and rational, and, perhaps most important, it does give good answers to the allocation and scheduling problems. Actual comparisons of this allocation procedure with others being used are necessary to demonstrate the degree of 'goodness' of this program.

The analytical models which were summarized earlier are felt to fall short of presenting useful answers to the shipyard planners. This is not because the solutions are not good, but rather because they do not apply to the exact problem which faces the shipyard planners- the multiproject, multiresource allocation problem. In addition,the solutions require great effort and time to obtain. Thus, as a practical tool for the user, the solution procedures would have to be computerized.

The disadvantages of Wiest's algorithm have already been pointed out. They can be summarized with the comment that Wiest's algorithm does not consider the dynamic nature of the workload at a shipyard. His algorithm would seem to be more applicable to a relatively static problem like the construction of a building or the manufacture of some product.

The empirical method that is currently being used in some shipyards also lacks flexibility and suffers in that it is really intended for the single project, multiresource case. In addition, it only tells the shop planner the number of men he must allocate, not the specific activities to which they must be assigned. Even with the addition of PERT/TIME to the empirical method, there are no provisions for handling the dependencies between projects which are created because of competition for scarce resources.

Lastly, scheduling with the modified empirical procedure is made on a weekly basis reducing the ability of shipyard planners to react quickly to those unforseen problems that occur.

The computer program presented in this thesis is very flexible. Because it allows for daily updating, shipyard planners can easily change schedules and resource allocations to adapt to current conditions and requirements. Its output is designed to contain all the information that a planner might need to answer the questions he faces daily. In addition, the input data requirements are simple enough so that litle time is required to obtain the answers he needs. Thus,the program provides answers to the planners in a timely manner.

Although written with the specific problem of the shipyard planners in mind, the computer program should be

useful for a host of other multiproject,multiresource allocation/scheduling problems.

A. A PROCEDURE TO ORDER NETWORK EVENT NODES TOPOLOGICALLY (Fulkerson [Ref.11])

1. Number the initial project event (node with no predecessor activities) with 1.

2. Delete all activities from the initial event (node) and search for events in the new network that are now initial events; number these 2,3,.. from top to bottom.

3. Repeat step 2 until the terminal project event (sink node) is numbered.

B. PROCEDURE TO CONVERT AN ACTIVITY-ON-NODE DIAGRAM INTO ACTIVITY-ON-ARC DIAGRAM

(From J.H.Cyr [Ref.12])

Definitions:

A merge node is that node which has two or more arcs incident into it.

Algorithm

I. Insert the Project Start Event

A. Beginning with the activity-on-node network, add a new node (call it a source node, which represents the 'Project Start ' event). Draw (directed) arcs from this source node to each other node of the network which has no predecessor.

II.Move Activities fron Nodes to Arcs

A. For each node which is not a merge node, move the label from the node to the associated arc incident into it (delete the label from a node once that label has been moved to an arc).

B. For each merge node:

1. Change each arc merging into the node to a dummy arc(the node is now a merge node for dummy arcs).

2. 'Split' each merge node into two unlabelled nodes (one of these will be a merge node) joined by an arc directed from the merge node to the second new node, and carrying the label of the original node. If the original merge node had diverging arcs, these must be placed at the second new node.

III.Insert Project Completion Event

A. Combine (superimpose) all nodes with no successor into a single unlabelled node called the 'Project Completion Event' (sink), which now becomes a merge node for all arcs incident into the original nodes.

IV. Eliminate Unnecessary Dummy Arcs

A. A dummy arc is unnecessary if it is the only arc incident from a node. Eliminate each unnecessary dummy arc by combining its two end nodes into a single node.

The resulting network is an 'Activity-on-Arc' project network. Each node represents a milestone or 'event'.

C. SUBROUTINE STOC

1. Purpose

The purpose of this subroutine is to select the activities that will have crew assignments, according to a Monte Carlo Method.

2. Description

Before calling subroutine STOC, the jobs that have become active for the day will have been placed on a list. Suppose that there are K activities on the list. A uniform distribution is assumed, so each activity has probability 1/K of being chosen.

Using a random number generator the first activity to be selected is obtained.It is taken out of the list. After this is done , there are only K-1 activities on the list and the probability of selection for each activity is recalculated to be 1/(K-1).

To keep track of the order in which the activities have been selected, a pointer is used to indicate the priority under which the activities on the original list have to be considered when the ASSCR and ASSNC subroutines are selecting the critical and noncritical activities, respectively, from their lists for the allocation of resources.

The random number generator will start from the same seed every time the program is used, if no change is made on the variable KX (the seed). Thus , care must be taken to change KX if Random Selection is being used to produce alternative schedules for the complete project duration. The seed can be any odd number, not greater than 5 digits.

D. SUBROUTINE BORROW

1. Purpose

To reassign resources among the critical jobs, when the resources available are not sufficient to satisfy all the active critical activities with at least a minimum crew.

2. Description

After the subroutine ASSCR has finished the resource assignment, the program tests to see if there are some active critical jobs with no resources assigned to them.If the answer is yes, the subroutine BORROW makes a list of all the active critical jobs that have crew sizes larger than their minimum crew sizes. These become the possible donors. A list is made for each type of resource.

Another list is made for the activities with no resources assigned,the possible acceptors. These jobs must be active. This list is also broken down according to the type of resource needed.

The subroutine BORROW determines whether or not an activity will become an actual donor and how many men it should release by examining the changes in slack times of both the prospective donor and acceptor. An attempt is made to prevent project delays whenever possible.

The process of searching for donors for each critical activity with insufficient resources continues as long there are acceptors and donors remaining.

E. SUBROUTINE EXHST

1. Purpose.

The purpose of the subroutine EXHST is to reassign resources to the active noncritical jobs when there are resources left unassigned after the subroutine ASSNC has performed its allocation to noncritical jobs.

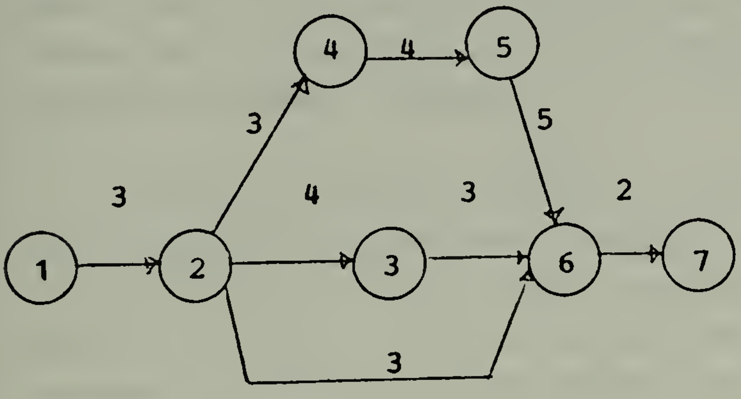
2. Description.

After the subroutine ASSNC has completed the crew assignment tc the noncritical jobs, EXHST tests to determine if there are still resources not used that can be assigned to augment the normal crew sizes for the noncritical jobs. If so then EXHST scans all the active noncritical jobs, determines the types of resources needed and assigns maximum crews where possible.

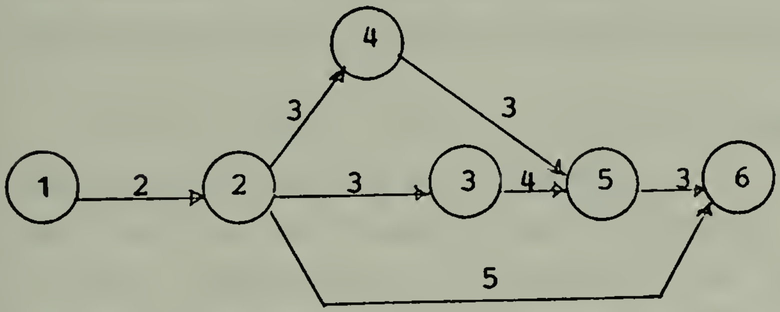
The procedure terminates when either the resources are exhausted or all the active noncritical jobs have been assigned a maximum crew size.

F. INTERPRETATION OF THE COMPUTER PROGRAM OUTPUT

As an example to illustrate how the program works, the following two networks, each representing a project will be used. The activity duration has been indicated in days for a normal crew size. The maximum and minimum crew sizes are those listed in the table following 'Status Of Activities', in the computer output.



Project 1



Project 2

Fig.9

Two computer outputs have been obtained. One is for day three where ample resources have been assumed available (20 for both types) ; the next output corresponds to day six, where the resources available are very restricted (3 of type 1 and 3 of type 2).

The output sample obtained for day 3 will be used as a reference.

Under the title 'Status of Projects' the printout of the node-arc matrices describing the projects involved, and used as input, will be followed by a list of the parameters that are usually calculated in a PERT network for each activity. These are from left to right: job, early start time (ES), early finish time (EF), latest finish time (LF), latest start time (LS) and the slack(S). This output is repeated for each project.

Under the heading 'Status of Activities', a table shows the activities that have become active for each project involved during the day considered. The variables D:duration of the activity in days and HH:man-days needed to complete the job have been added to the other variables already described.
The number of critical and noncritical jobs for each project are presented in a summary below each table.

On the next page, a matrix shows all the updated parameters used for each activity.
Min, Nor and Max give the minimum,normal and maximum crew sizes, respectively; CD indicates the type of resource used; SW tells whether the job is active (SW=1) or not; CR gives the crew assigned to the job; and I-J describes the job in terms of the beginning and finishing nodes.

The next page of output presents information about the critical jobs. The first table shows the ordering of the critical jobs, and the second table shows the resource assignments for these jobs. When the variable, HH, has the value 0(for example, as in jobs 1-2 of projects 1 and 2) there is no more work to be done on those jobs. Hence, CR, the crew assigned, should also be 0. When the variable,

Code, has a value 0 the activity is finished. In the example, activities 1-2 of projects 1 and 2 are finished (which can also be seen on the tables under the heading 'Status of Activities'). AV(1) and AV(2) indicate the resources of type 1 and type 2 that are still available (17 and 17 respectively).

A final page is devoted to the noncritical activities, The same information as described for the critical activities is presented. In this example, since the resources available were ample, there are 15 units of resource 1 and 8 of resource 2 still available after assigning men to all critical and noncritical jobs. The program then goes through the Subroutine EXHST, which tries to use all the resources left. The result is seen in the table, Revised Crew Assignment For Noncritical Jobs, where the crew for every job has been increased to the maximum crew size.

The output for day six is described for the case where the resources available have been reduced to 3 men for each of the type 1 and type 2 resources and the active critical jobs need only resources of type 1.

The structure of the tables, presented under the headings, 'Status Of Projects', ,'Status Of Activities' and 'Updated Projects' are identical to those tables in the previous example.

In the table,'Crew Assigned to Critical Jobs' one can observe that job(4-5) of project 1 has been assigned no resources and it needs 9 man-days (HH=9) to be completed. One can also see that this activity needs type 1 resources and AV(1)=0 (no type 1 resources are available). It is also evident that there is a surplus of units of type 2 resource (AV(2)=3).

In the next table, 'Revised Crew Assignment For Critical Jobs' one can see that the program has found resources for job(4-5) of project 1 by using the subroutine BORROW.

One man was borrowed from activity(3-5) in project 2 and reassigned to activity(4-5) of project 1, which can now get underway with a minimum crew size (CR=1).

Since there are still type 2 resources available, the program assigns the remaining three men to activity(2-6) in project 2. This can be seen by looking at the table 'Crew Assigned To Noncritical Jobs'.

```
                    DATE :    3
            STATUS    OF     PROJECTS

ECHO   CHECK  INPUT   MATRIX


0    3    0    0    0    0    0
0    0    4    3    0    3    0
0    0    0    0    0    3    0
0    0    0    0    4    0    0
0    0    0    0    0    5    0
0    0    0    0    0    0    2
0    0    0    0    0    0    0
             PROJECT    NUMBER     1
JOB    ES(I)   EF    LF(J)   LS        S
1- 2     1       4      4       1      0
2- 3     4       8     13       9      5
2- 4     4       7      7       4      0
4- 5     7      11     11       7      0
2- 6     4       7     16      13      9
3- 6     8      11     16      13      5
5- 6    11      16     16      11      0
6- 7    16      18     18      16      0
ECHO   CHECK  INPUT   MATRIX


0    2    C    0    0    0
0    0    3    3    0    5
0    0    0    0    4    0
0    0    0    0    3    0
0    0    0    0    0    3
0    0    0    0    0    0
             PROJECT    NUMBER     2
JOB    ES(I)   EF    LF(J)   LS        S
1- 2     1       3      3       1      0
2- 3     3       6      6       3      0
2- 4     3       6      7       4      1
3- 5     6      10     10       6      0
4- 5     6       9     10       7      1
2- 6     3       8     13       8      5
5- 6    10      13     13      10      0
```

```
               STATUS    OF   ACTIVITIES

                PROJECT    NUMBER     1
              ACTIVE AND FINISHED JOBS

        JOB       D    EF    LF    LS     S    HH

        1- 2     IS   FINISHED
        2- 3      4     7    13     9     6    12
        2- 4      3     6     7     4     0     6
        2- 6      3     6    16    13    10     9
         ACTIVE-CRITICAL JOBS =  2
        ACTIVE-NONCRITICAL JOBS =  2

                PROJECT    NUMBER     2
              ACTIVE AND FINISHED JOBS

        JOB       D    EF    LF    LS     S    HH

        1- 2     IS   FINISHED
        2- 3      3     6     6     3     0     6
        2- 4      3     6     7     4     1     6
        2- 6      3     6    13     8     7    10
         ACTIVE-CRITICAL JOBS =  2
        ACTIVE-NONCRITICAL JOBS =  2
```

UPDATED PROJECTS

I-	J	ES	D	EF	LF	LS	S	HH	MIN	NOR	MAX	CD	SW	CR
1	2	3	3	3	4	1	0	0	1	2	3	0	1	3
2	3	3	4	7	13	9	6	12	2	3	4	2	1	0
2	4	3	3	6	7	4	0	6	1	2	3	1	1	0
4	5	6	4	11	11	7	0	12	2	3	4	1	0	0
2	6	3	3	6	16	13	10	9	2	3	4	2	1	0
3	6	7	3	11	16	13	5	6	1	2	3	2	0	0
5	6	11	5	16	16	11	0	10	1	2	3	1	0	0
6	7	16	2	18	18	16	0	4	1	2	3	2	0	0
1	2	3	2	3	3	1	0	0	1	2	3	0	1	1
2	3	3	3	6	6	3	0	6	1	2	3	2	1	0
2	4	3	3	6	7	4	1	6	1	2	3	1	1	0
3	5	6	4	10	10	6	0	12	2	3	4	2	0	0
4	5	6	3	9	10	7	1	6	1	2	3	1	0	0
2	6	3	3	6	13	8	7	10	2	3	4	2	1	0
5	6	10	3	13	13	10	0	6	1	2	3	1	0	0

```
ACTIVE  CRITICAL   JOBS IN  PRIORITY  FOR  ASSIGNMENT

     PROJECT  JOB    ES    D   EF   LF   LS   HH


        2     1- 2    3    2    3    3    1    0
        1     1- 2    3    3    3    4    1    0
        1     2- 4    3    3    6    7    4    6
        2     2- 3    3    3    6    6    3    6
        CREW   ASSIGNED   TO  CRITICAL  JOBS
     PROJECT  JOB    CR   HH  CODE AV(1) AV(2)


        2     1- 2    0    0    0   17   17
        1     1- 2    0    0    0   17   17
        1     2- 4    3    6    1   17   17
        2     2- 3    3    6    2   17   17
```

```
ACTIVE NONCRITICAL JOBS IN  PRIORITY  FOR  ASSIGNMENT

     PROJECT   JCB     ES     D    EF    LF    LS    HH


        2      2- 4     3     3     6     7     4     6
        1      2- 3     3     4     7    13     9    12
        2      2- 6     3     3     6    13     8    10
        1      2- 6     3     3     6    16    13     9
          CREW   ASSIGNED  TO NONCRITICAL   JOBS
     PROJECT   JOB     CR     HH   CODE AV(1) AV(2)


        2      2- 4     2      6     1    15    17
        1      2- 3     3     12     2    15    14
        2      2- 6     3     10     2    15    11
        1      2- 6     3      9     2    15     8
    REVISED  CREW  ASSIGNMENT  FOR  NON CRITICAL  JOBS


      PROJECT   JOB     ES     D    EF    LF     S    CD    CR


         2      2- 4     3     3     6     7     1     1     3
         1      2- 3     3     4     7    13     6     2     4
         2      2- 6     3     3     6    13     7     2     4
         1      2- 6     3     3     6    16    10     2     4
                END OF SCHEDULE FOR THE DAY
```

```
                    DATE :    6
              STATUS    OF     PROJECTS

ECHO   CHECK  INPUT   MATRIX


0    3    0    0    0    0    0
0    0    4    3    0    3    0
0    0    0    0    0    3    0
0    0    0    0    4    0    0
0    0    0    0    0    5    0
0    0    0    0    0    0    2
0    0    0    0    0    0    0
               PROJECT    NUMBER      1
JOB    ES(I)   EF    LF(J)   LS        S
1- 2     1      4      4      1        0
2- 3     4      8     13      9        5
2- 4     4      7      7      4        0
4- 5     7     11     11      7        0
2- 6     4      7     16     13        9
3- 6     8     11     16     13        5
5- 6    11     16     16     11        0
6- 7    16     18     18     16        0
ECHO   CHECK  INPUT   MATRIX


0    2    0    0    0    0
0    0    3    3    0    5
0    0    0    0    4    0
0    0    0    0    3    0
0    0    0    0    0    3
0    0    0    0    0    0
               PROJECT    NUMBER      2
JOB    ES(I)   EF    LF(J)   LS        S
1- 2     1      3      3      1        0
2- 3     3      6      6      3        0
2- 4     3      6      7      4        1
3- 5     6     10     10      6        0
4- 5     6      9     10      7        1
2- 6     3      8     13      8        5
5- 6    10     13     13     10        0
```

```
                STATUS    OF    ACTIVITIES

                 PROJECT    NUMBER     1
               ACTIVE AND FINISHED JOBS

   JOB       D    EF     LF     LS      S     HH

   1- 2     IS   FINISHED
   2- 3      2     8     13      9      5      6
   2- 4     IS   FINISHED
   4- 5      3    10     11      7      0      9
   2- 6      2     8     16     13      8      6
  ACTIVE-CRITICAL JOBS =   3
 ACTIVE-NONCRITICAL JOBS =   2

                 PROJECT    NUMBER     2
               ACTIVE AND FINISHED JOBS

   JOB       D    EF     LF     LS      S     HH

   1- 2     IS   FINISHED
   2- 3     IS   FINISHED
   2- 4      1     7      7      4      0      2
   3- 5      4    10     10      6      0     12
   2- 6      3     9     13      8      4     10
  ACTIVE-CRITICAL JOBS =   4
 ACTIVE-NONCRITICAL JOBS =   1
```

UPDATED PROJECTS

I-	J	ES	D	EF	LF	LS	S	HH	MIN	NOR	MAX	CD	SW	CR
1	2	6	3	6	4	1	0	0	1	2	3	0	1	0
2	3	6	2	8	13	9	5	6	2	3	4	2	1	3
2	4	6	3	6	7	4	0	0	1	2	3	0	1	0
4	5	7	3	10	11	7	0	9	2	3	4	1	1	3
2	6	6	2	8	16	13	8	6	2	3	4	2	1	0
3	6	8	3	11	16	13	5	6	1	2	3	2	0	0
5	6	10	5	16	16	11	0	10	1	2	3	1	0	0
6	7	16	2	18	18	16	0	4	1	2	3	2	0	0
1	2	6	2	6	3	1	0	0	1	2	3	0	1	0
2	3	6	3	6	6	3	0	0	1	2	3	0	1	0
2	4	6	1	7	7	4	0	2	1	2	3	1	1	0
3	5	6	4	10	10	6	0	12	2	3	4	2	1	0
4	5	7	3	9	10	7	1	6	1	2	3	1	0	0
2	6	6	3	9	13	8	4	10	2	3	4	2	1	0
5	6	10	3	13	13	10	0	6	1	2	3	1	0	0

```
ACTIVE  CRITICAL   JOBS IN  PRIORITY  FOR  ASSIGNMENT

     PROJECT  JOB     ES     D    EF    LF    LS    HH


        2     2- 4     6     1     7     7     4     2
        2     1- 2     6     2     6     3     1     0
        1     1- 2     6     3     6     4     1     0
        1     2- 4     6     3     6     7     4     0
        2     2- 3     6     3     6     6     3     0
        2     3- 5     6     4    10    10     6    12
        1     4- 5     7     3    10    11     7     9
         CREW    ASSIGNED    TO   CRITICAL   JOBS
     PROJECT  JOB     CR    HH   CODE  AV(1)  AV(2)


        2     2- 4     2     2     1     0     0
        2     1- 2     0     0     0     0     0
        1     1- 2     0     0     0     0     0
        1     2- 4     0     0     0     0     0
        2     2- 3     0     0     0     0     0
        2     3- 5     3    12     2     0     0
        1     4- 5     1     9     1     0     0
                 END OF SCHEDULE FOR THE DAY
```

```
COMPUTER   PROGRAM
C     COMPUTER PROGRAM FOR ALLOCATING RESOURCES IN A MULTIPROJECT CASE
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 HH(7,7),LST/15/,MIJ/1/,L/0/,K/1/,LUT(2),T
      INTEGER*2 X(15),Z(15)
      INTEGER*2 CMIN(7,7),CNOR(7,7),CMAX(7,7),CODE(7,7),SW(7,7),CR(7,7)
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 A(195),C(195),IB(15),JB(15),CX(15),DD(30),DS,SK
      INTEGER*2 LU/12/,CTE,AV(2),IPT(15)
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
      READ(5,1)AV(1),AV(2),KST,KV,LUT(1),LUT(2),NP,T,NRUNS
    1 FORMAT(9I4)
      WRITE(6,55)T
   55 FORMAT('1',40X,'DATE :',I4)
      WRITE(6,45)
   45 FORMAT('0',31X,'STATUS   OF    PROJECTS'/)
      WRITE(6,161)
  161 FORMAT('0',23X,'ECHO  CHECK INPUT  MATRIX'//)
  705 IF(L.NE.0)GO TO 706
   28 GO TO(15,16),K
   15 LK=LUT(1)
      GO TO 2
   16 LK=LUT(2)
      WRITE(6,161)
    2 DO 3 I=1,LK
      GO TO(18,19),K
   18 READ(5,5)(D(I,J),J=1,LK)
      WRITE(6,40)(D(I,J),J=1,LK)
   40 FORMAT('0',20X,7I4)
    5 FORMAT(7I4)
      GO TO 3
   19 READ(5,29)(D(I,J),J=1,LK)
      WRITE(6,41)(D(I,J),J=1,LK)
   41 FORMAT('0',20X,6I4)
   29 FORMAT(6I4)
    3 CONTINUE
   17 DO 4 J=2,LK
      DO 4 I=1,J
      IF(D(I,J).EQ.0)GO TO 4
      IF(MIJ.EQ.2)GO TO 6
      READ(5,11)HH(I,J),CMIN(I,J),CNOR(I,J),CMAX(I,J),CODE(I,J),SW(I,J),
     1CR(I,J)
   11 FORMAT(7I4)
      GO TO 4
  706 IF(KST.EQ.0)GO TO 28
      DO 51 I=1,LST
      I1=I+LST
      I2=I1+LST
```

```
      I3=I2+LST
      I4=I3+LST
      I5=I4+LST
      I6=I5+LST
      I7=I6+LST
      I8=I7+LST
      I9=I8+LST
      I10=I9+LST
      I11=I10+LST
      I12=I11+LST
      X(I)=IB(I)
      Z(I)=JB(I)
      A(I)=C(I)
      A(I1)=C(I1)
      A(I2)=C(I2)
      A(I3)=C(I3)
      A(I4)=C(I4)
      A(I5)=C(I5)
      A(I6)=C(I6)
      A(I7)=C(I7)
      A(I8)=C(I8)
      A(I9)=C(I9)
      A(I10)=C(I10)
      A(I11)=C(I11)
      A(I12)=C(I12)
   51 CCNTINUE
      GC TO 52
    6 L=L+1
      A(L)=ES(I)
      X(L)=I
      Z(L)=J
      M=L+LST
      A(M)=D(I,J)
      M=M+LST
      A(M)=EF(I,J)
      M=M+LST
      A(M)=LF(J)
      M=M+LST
      A(M)=LS(I,J)
      M=M+LST
      A(M)=S(I,J)
      M=M+LST
      A(M)=HH(I,J)
      M=M+LST
      A(M)=CMIN(I,J)
      M=M+LST
      A(M)=CNOR(I,J)
      M=M+LST
```

```
      A(M)=CMAX(I,J)
      M=M+LST
      A(M)=CODE(I,J)
      M=M+LST
      A(M)=SW(I,J)
      M=M+LST
      A(M)=CR(I,J)
    4 CONTINUE
      IF(MIJ.EQ.2)GO TO 8
      GO TO(12,13),K
   12 CALL BWFWD(LK,K)
   14 MIJ=2
      GO TO 17
   13 CALL BWFWD(LK,K)
      GO TO 14
    8 K=K+1
      IF(K.GT.NP)GO TO 7
      MIJ=1
      GO TO 28
    7 CONTINUE
   52 K=1
      DO 140 I=1,30
      DD(I)=0
  140 CONTINUE
      WRITE(6,147)
  147 FORMAT('0',31X,'STATUS     OF    ACTIVITIES'///)
  145 GO TO(143,144),K
  143 LK=LUT(1)+1
      LI=1
      WRITE(6,148)K
  148 FORMAT(' ',32X,'PROJECT    NUMBER',3X,I2//,
     131X,'ACTIVE AND FINISHED JOBS'///,
     225X,'JOB      D     EF     LF     LS      S     HH'/)
      GO TO 146
  144 LK=LUT(2)+1+LK
      LI=LI+LUT(1)+1
      WRITE(6,148)K
  146 CALL UPDAT(LK,T,LI,K)
      K=K+1
      IF(K.LE.NP)GO TO 145
C       LU   IS   PARAMETROS  USADOS EN   EL   NET.
      K=0
      WRITE(6,162)
  162 FCRMAT('1',35X,'UPDATED     PROJECTS'///)
      WRITE(6,163)
  163 FORMAT('0',24X,'NO     ES     D     EF     LF     LS      S      HH      MI
     1N    NOR     MAX    CD     SW     CR     I  -  J'//)
      DO 840 I=1,15
```

```
      I1=I+15
      I2=I1+15
      I3=I2+15
      I4=I3+15
      I5=I4+15
      I6=I5+15
      I7=I6+15
      I8=I7+15
      I9=I8+15
      I10=I9+15
      I11=I10+15
      CX(I)=0
      I12=I11+15
      WRITE(6,142)I,A(I),A(I1),A(I2),A(I3),A(I4),A(I5),A(I6),A(I7),A(I8)
     1,A(I9),A(I10),A(I11),A(I12),X(I),Z(I)
  142 FORMAT('0',20X,16I6)
  840 CONTINUE
      DO 200 I=1,LST
      I5=I+5*LST
      I11=I+11*LST
      IF(A(I11).NE.1)GO TO 200
      IF(A(I5).GT.0)GO TO 200
      K=K+1
      I1=I+LST
      K1=K+NCR
      DD(K)=A(I)
      DD(K1)=A(I1)
      CX(K)=I
  200 CONTINUE
C     ORDER   BY  ASCENDING   ES
      MM=NCR-1
      KT=1
  203 DC 201 I=1,MM
      J=I+1
      I1=I+NCR
      J1=I1+1
      IF(DD(J).LT.DD(I))GO TO 208
      IF(DD(J).NE.DD(I))GO TO 201
      IF(KT.LE.1)GO TO 201
C     ORDER BY  ASCENDING  D
      IF(DD(J1).GE.DD(I1))GO TO 201
  208 CTE=CX(J)
      CX(J)=CX(I)
      CX(I)=CTE
      CTE=DD(J)
      DD(J)=DD(I)
      DD(I)=CTE
      CTE=DD(J1)
```

```
      DD(J1)=DD(I1)
      DD(I1)=CTE
  201 CONTINUE
      MM=MM-1
      IF(MM.LE.1)GO TO 205
      GO TO 203
  205 KT=KT+1
      IF(KT.GT.2)GO TO 207
      MM=NCR-1
      GO TO 203
  207 WRITE(6,206)
  206 FORMAT('1',11X,'ACTIVE  CRITICAL   JOBS IN  PRIORITY  FOR  ASSIGNM
     1ENT'///,
     217X,'PROJECT  JOB     ES     D    EF    LF    LS    HH'//)
  202 CALL MODUL(LU,NCR,LST)
      IF(KST.NE.1)GO TO 209
      CALL STOC(NCR)
  209 CALL ASSCR(NCR,LST,AV(1),AV(2),KD1,KD2,KD3)
      IF(KD1.NE.-9)GO TO 204
      CALL BORROW(NCR,LST)
      WRITE(6,160)
  160 FORMAT('0',19X,'REVISED  CREW  ASSIGNMENT  FOR  CRITICAL  JOBS'///
     1)
      WRITE(6,149)
  149 FORMAT('0',17X,'PROJECT  JOB    ES    D    EF    LF    LS    CD    CR'/
     1/)
      DO 150 I=1,NCR
      I1=I+15
      I2=I1+15
      I3=I2+15
      I4=I3+15
      I10=I+150
      I12=I+180
      IF(CX(I).LT.9)GO TO 164
      KR=2
      GO TO 165
  164 KR=1
  165 WRITE(6,151)KR,IB(I),JB(I),C(I),C(I1),C(I2),C(I3),C(I4),C(I10),C(I
     112)
  151 FORMAT('0',19X,I2,I6,'-',I2,7I5)
  150 CONTINUE
  204 IF((KD2.NE.0).OR.(KD3.NE.0))GO TO 547
      GO TO 560
  547 K=0
      IF(NOCR.EQ.0)GO TO 560
      DO 548 I=1,LST
      I1=I+LST
      I5=I+5*LST
```

```
      I11=I+11*LST
      IF(A(I11).NE.1)GO TO 548
      IF(A(I5).EQ.0)GO TO 548
      K=K+1
      DD(K)=A(I)
      CX(K)=I
      K1=K+NOCR
      K5=K1+NOCR
      DD(K1)=A(I1)
      DD(K5)=A(I5)
  548 CONTINUE
C     ORDER IN ASCENDING ES, D   , S
      MM=NOCR-1
      KT=1
  549 DO 550 I=1,MM
      J=I+1
      I1=I+NOCR
      I5=I1+NOCR
      J1=I1+1
      J5=I5+1
      IF(DD(J).LT.DD(I))GO TO 551
      IF(DD(J).NE.DD(I))GO TO 550
C     ORDER  BY  ASCENDING  S
      IF(KT.LE.1)GO TO 550
      IF(DD(J5).LT.DD(I5))GO TO 551
      IF(DD(J5).NE.DD(I5))GO TO 550
      IF(KT.LE.2)GO TO 550
C     ORDER  BY   D
      IF(DD(J1).GE.DD(I1))GO TO 550
  551 CTE=CX(J)
      CX(J)=CX(I)
      CX(I)=CTE
      CTE=DD(J)
      DD(J)=DD(I)
      DD(I)=CTE
      CTE=DD(J5)
      DD(J5)=DD(I5)
      DD(I5)=CTE
      CTE=DD(J1)
      DD(J1)=DD(I1)
      DD(I1)=CTE
  550 CONTINUE
      MM=MM-1
      IF(MM.LE.1)GO TO 553
      GO TO 549
  553 KT=KT+1
      IF(KT.GT.3)GO TO 554
      MM=NOCR-1
```

```
      GO TO 549
  554 WRITE(6,555)
  555 FORMAT('1',11X,'ACTIVE NONCRITICAL JOBS IN  PRIORITY  FOR  ASSIGNM
     1ENT'///,
     217X,'PROJECT  JOB    ES     D    EF    LF    LS    HH'//)
  552 CALL MODUL(LU,NOCR,LST)
      CALL ASSNC(NOCR,LST,AV(1),AV(2),KD1,KD2,KD3)
      IF((KD2.NE.0).OR.(KD3.NE.0))GO TO 561
      GO TO 560
  561 CALL EXHST(KV,NOCR,LST)
  560 CONTINUE
      DO 700 I=1,LST
      I6=I+6*LST
      IF(C(I6).GT.0)GO TO 707
  700 CONTINUE
      NRUNS=NRUNS-1
      IF(NRUNS.LT.0)GO TO 708
  709 L=1
      T=T+1
      GO TO 705
  707 IF(KST.NE.0)GO TO 709
      WRITE(6,710)
  710 FORMAT('0',27X,'END OF SCHEDULE FOR THE DAY')
      GO TO 711
  708 WRITE(6,712)
  712 FORMAT('0',25X,'END  OF  PROJECT')
  711 STOP
      END
      SUBROUTINE BWFWD(LT,KPR)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 X(15),Z(15)
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 A(195),C(195),IB(15),JB(15),CX(15),DD(30),DS,SK
      INTEGER*2 T,IPT(15),AV(2)
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
      ES(1)=1
      DO 20 J=2,LT
      N=0
      DO 21 I=1,J
      IF(D(I,J).EQ.0)GO TO 21
      EF(I,J)=ES(I)+D(I,J)
      N=N+1
      C(N)=EF(I,J)
   21 CONTINUE
      IF(N.GT.1)GO TO 30
      ES(J)=C(N)
      GO TO 32
   30 MAX=C(1)
```

```
      DO 22 K=2,N
      IF(MAX.GT.C(K))GO TO 22
      MAX=C(K)
   22 CONTINUE
      ES(J)=MAX
   32 CONTINUE
      LF(J)=ES(J)
   20 CONTINUE
      LU=LT-1
      DO 24 II=1,LU
      I=LT-II
      N=0
      DO 25 J=1,LT
      IF(D(I,J).EQ.0)GO TO 25
      LS(I,J)=LF(J)-D(I,J)
      N=N+1
      C(N)=LS(I,J)
   25 CONTINUE
      IF(N.GT.1)GO TO 33
      LF(I)=C(N)
      GC TO 24
   33 MIN=C(1)
      DC 26 K=2,N
      IF(MIN.LT.C(K))GO TO 26
      MIN=C(K)
   26 CONTINUE
      LF(I)=MIN
   24 CONTINUE
      WRITE(6,43)KPR
   43 FORMAT('0',32X,'PROJECT   NUMBER',3X,I2)
      WRITE(6,44)
   44 FORMAT('0',23X,'JOB   ES(I)   EF   LF(J)   LS     S')
      DO 27 J=2,LT
      DC 27 I=1,J
      IF(D(I,J).EQ.0)GO TO 27
      S(I,J)=LF(J)-EF(I,J)
      WRITE(6,42)I,J,ES(I),EF(I,J),LF(J),LS(I,J),S(I,J)
   42 FORMAT('0',20X,I4,'-',I2,I5,4I6)
   27 CONTINUE
      RETURN
      END
      SUBROUTINE UPDAT(LT,T,LI,K)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 X(15),Z(15)
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 A(195),C(195),IB(15),JB(15),CX(15),DD(30),DS,SK
      INTEGER*2 T,IPT(15),AV(2)
      COMMCN A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
```

```
      NCK=0
      NOK=0
      DO 100 I=LI,LT
      I1=I+15
      I2=I+30
      I3=I+45
      I4=I+60
      I5=I+75
      I6=I+90
      I8=I+120
      I11=I+165
      I12=I+180
      IF(A(I11).EQ.0)GO TO 100
      CH=A(I6)
      CH=CH-A(I12)
      IF(CH.EQ.0)GO TO 101
      A(I6)=A(I6)-A(I12)
      INT=A(I6)/A(I8)
      ANF=A(I8)
      AHH=A(I6)
      AXF=AHH/ANF
      ADL=AXF-INT
      IF(ADL.GE..5)GO TO 102
      A(I1)=INT
      GC TO 103
  102 A(I1)=INT+1
  103 LEV=DD(I)-A(I1)
      IF(LEV.LT.2)GO TO 104
      WRITE(6,110)IB(I),JB(I)
  110 FORMAT('0',20X,'ACTIVITY',I4,'-',I2,'IS AHEAD 1 DAY, PERFORM FWD')
  104 IF(A(I).GE.T)GO TO 105
      A(I)=T
  105 A(I2)=A(I)+A(I1)
      IF(A(I5).EQ.0)GO TO 106
      A(I5)=A(I3)-A(I2)
  106 DD(I)=A(I1)
      GO TO 107
  101 A(I2)=T
      A(I)=T
      A(I6)=CH
      IF(A(I5).LE.0)GO TO 107
  108 A(I5)=A(I3)-A(I2)
C     UPDATING   IS   FINISHED
  107 CCNTINUE
  100 CONTINUE
C     SET  SWITCHES   ROUTINE  BEGINS
      DO 150 L=LI,LT
      L1=L+15
```

```
      L2=L+30
      L3=L+45
      L4=L3+15
      L5=L+75
      L6=L+90
      L10=L+150
      L11=L+165
      IF(A(L6).EQ.0)GO TO 151
      IF(A(L11).EQ.1)GO TO 152
C     UPDATE  ES  IN CASE  PRECEDING ACTIV.  FINISHED  EARLIER
      I=0
      IN=X(L)
      DO 153 M=LI,LT
      IF(Z(M).NE.IN)GO TO 153
      I=I+1
      M2=M+30
      CX(I)=A(M2)
  153 CONTINUE
      IF(I-1)161,162,155
  161 A(L)=T
      GO TO 156
  162 A(L)=CX(I)
      GO TO 156
  155 MAX=CX(1)
      DO 154 J=2,I
      IF(MAX.GT.CX(J))GO TO 154
      MAX=CX(J)
  154 CONTINUE
      A(L)=MAX
  156 IF(A(L).GT.T)GO TO 157
      A(L11)=1
      A(L2)=A(L)+A(L1)
  152 IF(A(L5).EQ.0)GO TO 158
      A(L5)=A(L3)-A(L2)
      IF(A(L5).EQ.0)GO TO 158
      NCK=NOK+1
      GO TO 160
  157 A(L11)=0
      GO TO 150
  158 NCK=NCK+1
      GO TO 160
  151 CONTINUE
      WRITE(6,170)X(L),Z(L)
  170 FORMAT('0',23X,I2,'-',I2,'   IS  FINISHED')
      A(L2)=T
      A(L10)=0
      NCK=NCK+1
      GO TO 150
```

```
160 WRITE(6,149)X(L),Z(L),A(L1),A(L2),A(L3),A(L4),A(L5),A(L6)
149 FORMAT('0',23X,I2,'-',I2,6I5)
150 CONTINUE
    WRITE(6,171)NCK,NOK
171 FORMAT('0',23X,'ACTIVE-CRITICAL JOBS =',I3/,
   123X,'ACTIVE-NONCRITICAL JOBS =',I3,///)
    IF(K.LE.1)GO TO 172
    NCR=NCR+NCK
    NOCR=NOCR+NOK
    GC TO 173
172 NCR=NCK
    NCCR=NOK
173 CONTINUE
    RETURN
    END
    SUBROUTINE MODUL(LU,M,LST)
    IMPLICIT INTEGER*2 (I-N)
    INTEGER*2 X(15),Z(15)
    INTEGER*2 A(195),C(195),JB(15),IB(15),CX(15),DD(30),DS,SK
    INTEGER*2 LU,LST,AV(2),IPT(15)
    INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
    COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
    LUK=LU+1
    DO 210 I=1,LUK
    DO 211 J=1,M
    KB=(I-1)*LST
    KA=CX(J)+KB
    KC=J+KB
    C(KC)=A(KA)
211 CONTINUE
210 CONTINUE
    DO 218 J=1,M
    KA=CX(J)
    IB(J)=X(KA)
    JB(J)=Z(KA)
218 CONTINUE
    DO 212 J=1,M
    J1=J+LST
    J2=J1+LST
    J3=J2+LST
    J4=J3+LST
    J5=J4+LST
    J6=J5+LST
    J12=J+12*LST
    IF(CX(J).LT.9)GO TO 216
    KR=2
    GO TO 217
216 KR=1
```

```
217 WRITE(6,213)KR,IB(J),JB(J),C(J),C(J1),C(J2),C(J3),C(J4),C(J6)
213 FORMAT('0',18X,I2,I6,'-',I2,6I5)
212 CONTINUE
    RETURN
    END
    SUBROUTINE ASSCR(N,LST,AVE,AVI,KD1,KD2,KD3)
    IMPLICIT INTEGER*2 (I-N)
    INTEGER*2 CX(15),IB(15),JB(15),DD(30),IPT(15)
    INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
    INTEGER*2 A(195),C(195),AV(2),AVE,AVI,DS,SK
    INTEGER*2 X(15),Z(15)
    COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
    WRITE(6,267)
267 FORMAT('0',20X,'CREW   ASSIGNED   TO  CRITICAL  JOBS'//,
   117X,'PROJECT   JOB    CR    HH  CODE AV(1) AV(2)'//)
    DO 250 J=1,N
    IF(KST.EQ.1) GO TO 280
    I=J
    GO TO 281
280 I=IPT(J)
281 I1=I+LST
    I5=I+5*LST
    I6=I5+LST
    I7=I6+LST
    I9=I+9*LST
    I10=I9+LST
    I11=I10+LST
    I12=I11+LST
    IF(C(I10).EQ.0)GO TO 270
    MM=C(I10)
    IF(C(I1).GT.1)GO TO 251
    ICR=C(I6)
    IF(ICR.GT.C(I7))GO TO 252
    ICR=C(I7)
    GO TO 260
251 DC=C(I1)-2
    IF(DC.NE.0)GO TO 261
    DC=1
261 ICR=C(I6)/DC
    ANF=C(I6)
    AHH=DC
    AXF=ANF/AHH
    ADL=AXF-ICR
    IF(ADL.LT..5)GO TO 252
    ICR=ICR+1
252 IF(ICR.LE.C(I9))GO TO 260
    ICR=C(I9)
260 GO TO(253,254),MM
```

```
253 AVAIL=AVE
    GO TO 255
254 AVAIL=AVI
255 IF(ICR.LE.AVAIL)GO TO 256
    ICR=AVAIL
256 GO TO(257,258),MM
257 AVE=AVE-ICR
    GO TO 259
270 C(I12)=0
    GO TO 250
258 AVI=AVI-ICR
259 C(I12)=ICR
    IF(C(I12).GT.0)GO TO 264
    KD1=-9
    GO TO 250
264 KD1=9
250 CONTINUE
    KD2=AVE
    KD3=AVI
    DO 265 I=1,N
    I1=I+15
    I2=I1+15
    I3=I2+15
    I4=I3+15
    I5=I4+15
    I6=I5+15
    I7=I6+15
    I8=I7+15
    I9=I8+15
    I10=I9+15
    I11=I10+15
    I12=I11+15
    IF(CX(I).LT.9)GO TO 268
    KR=2
    GO TO 269
268 KR=1
269 WRITE(6,266)KR,IB(I),JB(I),C(I12),C(I6),C(I10),AVE,AVI
266 FORMAT('0',18X,I2,I6,'-',I2,5I5)
265 CONTINUE
    RETURN
    END
    SUBROUTINE ASSNC(N,LST,AVE,AVI,KD1,KD2,KD3)
    IMPLICIT INTEGER*2 (I-N)
    INTEGER*2 A(195),C(195),AV(2),AVE,AVI,DS,SK
    INTEGER*2 CX(15),IB(15),JB(15),DD(30)
    INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
    INTEGER*2 IPT(15)
    INTEGER*2 X(15),Z(15)
```

```
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
      WRITE(6,416)
  416 FORMAT('0',20X,'CREW  ASSIGNED  TO NONCRITICAL  JOBS'//,
     117X,'PROJECT  JOB    CR   HH  CODE AV(1) AV(2)'//)
      DO 400 I=1,N
      I1=I+LST
      I6=I+6*LST
      I7=I6+LST
      I8=I7+LST
      I9=I8+LST
      I10=I9+LST
      I12=I+12*LST
      IF(C(I10).EQ.0)GO TO 409
      KO=C(I10)
      GC TO(401,402),KO
  401 AVAIL=AVE
      GO TO 411
  402 AVAIL=AVI
  411 IF(C(I7).GT.AVAIL)GO TO 420
      IF(C(I1).NE.0)GO TO 407
      C(I1)=1
  407 ICR=C(I6)/C(I1)
      ANF=C(I6)
      AHH=C(I1)
      AXF=ANF/AHH
      ADL=AXF-ICR
      IF(ADL.LT..5)GO TO 408
      ICR=ICR+1
  408 IF(ICR.LE.C(I7))GO TO 409
      IF(ICR.LT.C(I8))GO TO 410
      ICR=C(I8)
      GC TO 410
  409 ICR=C(I6)
  410 GO TO(403,404),KO
  403 AVAIL=AVE
      GO TO 412
  404 AVAIL=AVI
  412 IF(ICR.LE.AVAIL)GO TO 413
      ICR=AVAIL
  413 GC TO(405,406),KO
  405 AVE=AVE-ICR
      GO TO 414
  406 AVI=AVI-ICR
  414 C(I12)=ICR
      I2=I+2*LST
      I3=I2+LST
      I4=I3+LST
      KD2=AVE
```

```
      KD3=AVI
      IF(CX(I).LT.9)GO TO 417
      KR=2
      GO TO 419
  417 KR=1
      GO TO 419
  420 C(I12)=0
      IF(CX(I).LT.9)GO TO 417
      KR=2
  419 WRITE(6,415)KR,IB(I),JB(I),C(I12),C(I6),C(I10),AVE,AVI
  415 FORMAT('0',18X,I2,I6,'-',I2,5I5)
  400 CONTINUE
      RETURN
      END
      SUBROUTINE BORROW(NCRK,LST)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 A(195),C(195),JB(15),IB(15),CX(15),DD(30),DS,SK
      INTEGER*2 DSD,CR,AV(2),IPT(15)
      INTEGER*2 K/0/,K1/1/,K2/0/,KI/1/,KJ/1/
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 X(15),Z(15)
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
C     LIST  ACTIVE  JOBS THAT  MAY  BECOME   DONORS
      DO 300 I=1,NCRK
      L=I+7*LST
      M=I+12*LST
      IF(C(M).LE.C(L))GO TO 300
      K=K+1
      N=I+10*LST
      IF(C(N).NE.1)GO TO 300
      K2=K2+1
C     LOWER  LIMIT  OF  CODE  2  LIST
  300 CONTINUE
      L=0
      M=0
      N=0
      DO 301 I=1,NCRK
      I7=I+7*LST
      I12=I+12*LST
      IF(C(I12).LE.C(I7))GO TO 301
      L=I+10*LST
      IF(C(L).EQ.1)GO TO 310
  311 K2=K2+1
      CX(K2)=L
      GC TO 313
 310  CX(K1)=L
      K1=K1+1
  313 CCNTINUE
```

```
  301 CONTINUE
      L=0
      K1=K1-1
      KJ=K1+1
      DO 302 I=1,NCRK
      I3=I+3*LST
      I6=I+6*LST
      I7=I+7*LST
      I8=I+8*LST
      I10=I+10*LST
      I12=I+12*LST
      IF(C(I12).GE.C(I7))GO TO 302
      I5=I+5*LST
C      TO   GET   CRIT.    JOBS   WITH   NO   CR
      IF(C(I5).GT.0)GO TO 302
      I11=I+11*LST
      IF(C(I11).NE.1)GO TO 302
      IF(C(I10).EQ.0)GO TO 302
C      HERE   IS   ONE   CRIT.   JOB   WITHOUT   CR
      KO=C(I10)
C      ITS   CODE   IS   C(I10)
      GO TO(314,315),KO
  314 INI=KI
      LI=K1
      GO TO 316
  315 INI=KJ
      LI=K2
  316 KD=0
      ICR=0
      IF(INI.GT.LI)GO TO 302
      DO 303 J=INI,LI
C     SCAN   LIST   OF   ACTIVE   JOBS   CODE1/2   FOR   DONOR
      KA=CX(J)
      JJ=KA-10*LST
      J3=KA-7*LST
      J5=KA-5*LST
      J6=J5+LST
      J7=J6+LST
      J8=KA-2*LST
      J12=KA+2*LST
      IF(C(J12).LE.C(J7))GO TO 333
      CR=C(J12)-1
      CALL DELTA(CR,C(J6),C(JJ),C(J8),C(J3),C(J5))
      IF(DS.LT.1)GO TO 317
      KD=KD+1
      ICR=ICR+1
      CALL DELTA(ICR,C(I6),C(I),C(I8),C(I3),C(I5))
C     TEST   ACEPTOR   DS   WITH   CR=1
```

```
      IF(SK.GE.C(I5))GO TO 342
      IF(C(J5).LT.C(I5))GO TO 318
      DSD=DS
      DO 306 L=1,DSD
C      ITERETION  NUMBER  OF  TIMES  DS  IS  OVER  0
      CR=CR-1
      CALL DELTA(CR,C(J6),C(JJ),C(J8),C(J3),C(J5))
      IF(DS.GE.0)GO TO 319
C      DONOR  CANNOT  GIVE  MORE
C      ACEPTOR  NEEDS  ANOTHER  DONOR
      GC TO 307
  319 KD=KD+1
      ICR=ICR+1
      IF(SK.LT.C(I5))GO TO 306
C      ACEPTOR  IS  OK
      GC TO 302
  306 CONTINUE
  318 CONTINUE
C      DONOR  CANNNOT  GIVE  MORE
      GO TO 332
  331 ICR=0
  332 GO TO(304,305),KO
  304 KI=KI+1
      GO TO 307
  305 KJ=KJ+1
      GO TO 307
  317 IF(DS.LT.0)GO TO 331
  330 IF(C(J5).LT.C(I5))GO TO 308
      KD=KD+1
      ICR=ICR+1
      CALL DELTA(ICR,C(I6),C(I),C(I8),C(I3),C(I5))
      IF(SK.GE.C(I5))GO TO 342
      GO TO(304,305),KO
  342 CONTINUE
      C(J12)=C(J12)-KD
      C(I12)=ICR
      GO TO 302
  308 ICR=0
C      TRY  ANOTHER  DONOR
      GO TO 332
  333 CCNTINUE
      GC TO(334,335),KO
  334 KI=KI+1
      GC TO 303
  335 KJ=KJ+1
      GO TO 303
  307 C(J12)=C(J12)-KD
      C(I12)=ICR
```

```
      KD=0
  303 CONTINUE
  302 CONTINUE
      RETURN
      END
      SUBROUTINE DELTA(CR2,HH2,ES2,CNOR2,LF2,SIJ)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 A(195),C(195),JB(15),IB(15),CX(15),DD(30),DS,SK
      INTEGER*2 CR2,HH2,ES2,CNOR2,SIJ,D2,EF2
      INTEGER*2 AV(2),IPT(15)
      INTEGER*2 X(15),Z(15)
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
      HA=HH2-CR2
      AHH=HA
      AXF=CNOR2
      ACR=CR2
      IF(HA.LE.CR2)GO TO 350
      D2=HA/CNOR2
      AD=AHH/AXF
      GO TO 351
  350 D2=HA/CR2
      AD=AHH/ACR
  351 AR=AD-D2
      IF(AR.LT..5)GO TO 352
      D2=D2+1
  352 EF2=ES2+1+D2
      SK=LF2-EF2
      DS=SK-SIJ
      RETURN
      END
      SUBROUTINE EXHST(KV,M,LST)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 A(195),C(195),DS,SK,AV(2),CR1
      INTEGER*2 CX(15),IB(15),JB(15),DD(30),IPT(15)
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 X(15),Z(15)
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
      WRITE(6,609)
  609 FORMAT('0',15X,'REVISED  CREW  ASSIGNMENT  FOR  NON CRITICAL  JOBS
     1'///)
      WRITE(6,610)
  610 FORMAT('0',17X,'PROJECT  JOB    ES    D   EF   LF    S    CD    CR'/
     1/)
      DO 600 J=1,KV
      IF(AV(J).LE.0)GO TO 600
      DO 601 I=1,M
      I6=I+6*LST
```

```
      I7=I6+LST
      I8=I7+LST
      I9=I8+LST
      I10=I9+LST
      I11=I10+LST
      I12=I11+LST
      I1=I+LST
      I2=I1+LST
      I3=I2+LST
      I5=I+5*LST
      IF(C(I10).NE.J)GO TO 601
      CR1=C(I6)
      IF(C(I12).GE.CR1)GO TO 601
      IF(CR1.GT.C(I9))GO TO 602
      IF(CR1.GE.C(I7))GO TO 603
      CR1=C(I7)
      GO TO 603
  602 CR1=C(I9)
  603 CTE=CR1-C(I12)
      IF(CTE.GT.AV(J))GO TO 604
      CR1=CTE
      GO TO 605
  604 CR1=AV(J)
  605 C(I12)=C(I12)+CR1
      AV(J)=AV(J)-CR1
      IF(CX(I).LT.9)GO TO 606
      KR=2
      GO TO 607
  606 KR=1
  607 WRITE(6,608)KR,IB(I),JB(I),C(I),C(I1),C(I2),C(I3),C(I5),C(I10),C(I
     112)
  608 FORMAT('0',19X,I2,I6,'-',I2,7I5)
  601 CONTINUE
  600 CONTINUE
      RETURN
      END
      SUBROUTINE STOC(NN)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 KX,IY
      INTEGER*2 A(195),C(195),JB(15),IB(15),CX(15),DD(30),DS,SK
      INTEGER*2 AV(2)
      INTEGER*2 ICX(15),DX(15),IPT(15)
      INTEGER*2 D(7,7),ES(7),EF(7,7),LS(7,7),LF(7),S(7,7)
      INTEGER*2 X(15),Z(15)
      COMMON A,AV,C,CX,D,DD,DS,EF,ES,IB,IPT,JB,LF,LS,NCR,NOCR,S,SK,X,Z
      DO 270 I=1,NN
      ICX(I)=I
      DX(I)=C(I)
```

```
  270 CONTINUE
      KX=65549
      N=NN
  199 DO 271 I=1,NN
      CALL RANDU(KX,IY,Y)
      KX=IY
      FINT=1.0/N
      IX=Y/FINT
      K=IX+1
      L=N-1
      IPT(I)=ICX(K)
      IF(L.LT.K)GO TO 278
      IF(L.EQ.1) GO TO 277
  273 DO 272 J=K,L
      J1=J+1
      ICX(J)=ICX(J1)
      DX(J)=DX(J1)
  272 CONTINUE
  278 N=N-1
      IF(K.NE.1)GO TO 271
      IF(N.LT.1)GO TO 280
      GO TO 271
  280 ICX(1)=ICX(2)
  271 CONTINUE
      GO TO 281
  277 M=1
      IF(K.NE.1)GO TO 274
      M=2
  274 IPT(NN)=ICX(M)
  281 CONTINUE
C     STOCH.  ORDERING  IS   FINISHED
      DO 279 I=1,NN
  279 CONTINUE
      RETURN
      END
```

LIST OF REFERENCES

1. O'Brien,J.J.,Scheduling Handbook,p. 494-495, McGraw-Hill Book Company,1969.

2. Davis,E.W.,"Resource Allocations in Project Network Models, a Survey," Jnl of Industrial Engineering, v.XVII, No.4,1966.

3. Burgess,A.R. and Killebrew,J.B.,"Variation in Activity Level on a Cyclic Arrow Diagram,"Jnl of Industrial Engineering, March-April 1962.

4. Wiest,J.D., Computer Models for the Scheduling of Large Projects, Ph.D. Dissertation, Carnegie Institute of Technology, September 1964.

5. Levy,F.K.,Thompson,G.L.,and Wiest,J.D.,"Multiship, Multishop, Workload Smoothing Program,"Naval Research Logistics Quarterly,v.9,No.1,March 1962.

6. Wiest,J.D.," A Heuristic Model for Scheduling Large Projects with Limited Resources,"Management Science, v.13,No.6,February 1967.

7. Shackelton,N.F.,Minimizing the Cost of Projects in Naval Shipyards,Ph.D. Thesis, Naval Postgraduate School, Monterey,1973.

8. Benders,J.F.,"Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numerische Mathematik,v.4,No.3,1962.

9. Bellman,R.K. and Dreyfus,S.S.,Applied Dynamic Programming,Princeton University Press,1962.

10. Moder,J.J. and Phillips,C.R.,Project Management with CPM and PERT,2d ed.,Van Nostrand Reinhold Co., 1970.

11. Fulkerson,D.R.,"Expected Critical Path Lengths in Pert Networks,"Operations Research,v.10, No.6,November-December 1962.

12. Cyr,J.H.,Procedure to Convert an Activity-on-Node Diagram into an Activity-on-Arc Diagram,Classnotes, Naval Postgraduate School, Monterey ,1973.

INITIAL DISTRIBUTION LIST

	No.Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Department Chairman Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
3. Comandancia en Jefe de la Armada Ministerio de Defensa Nacional Santiago, Chile	2
4. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
5. Asst Professor F.R.Richards, Code 55Rh Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6. Assoc Professor A.W.McMasters,Code 55Mg Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1

7. Lcdr. J.H.Cyr, Code 030 1
Department of Operations Research and
Administrative Sciences
Naval Postgraduate School
Monterey,California 93940

8. Cdr. Enrique A. Medina,Chilean Navy 1
Direccion General del Personal de la Armada
Correo Naval, Valparaiso, CHILE